HAL open science

Equivalence Between Model-Checking Flat Counter Systems and Presburger Arithmetic

Stéphane Demri, Amit K Kumar Dhar, Arnaud Sangnier

▶ **To cite this version:**

Stéphane Demri, Amit K Kumar Dhar, Arnaud Sangnier. Equivalence Between Model-Checking Flat Counter Systems and Presburger Arithmetic. Theoretical Computer Science, 2017, 735, pp.2-23. 10.1016/j.tcs.2017.07.007 . hal-03192244

HAL Id: hal-03192244

https://hal.science/hal-03192244v1

Submitted on 7 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Equivalence Between Model-Checking Flat Counter Systems and Presburger Arithmetic*

Stéphane Demri[1], Amit Kumar Dhar[2], and Arnaud Sangnier[3]

[1] LSV, CNRS & ENS Paris-Saclay, France
[2] Indian Institute of Information Technology, Allahabad, India
[3] IRIF, Univ Paris Diderot, France

Abstract. We show that model-checking flat counter systems with the branching-time temporal logic CTL* extended with arithmetical constraints on counter values has the same worst-case complexity as the satisfiability problem for Presburger arithmetic. The lower bound already holds with strong restrictions: the logical language uses only the temporal operator EF and no arithmetical constraints, and the guards on the transitions are made of linear constraints. This work complements our understanding of model-checking flat counter systems with linear-time temporal logics, such as LTL, for which the problem is already known to be (only) NP-complete with guards restricted to the linear fragment.

1 Introduction

Branching-time temporal logics for counter systems. At first glance, model-checking counter systems with temporal logics seems a hopeless enterprise since even the control-state reachability problem for Minsky machines is known to be undecidable [34]. However, we are also aware that decidability can be regained by considering some syntactical or semantical restrictions on the systems. As a matter of fact, many subclasses of counter systems admit a decidable reachability problem and even better, sometimes the reachability sets are definable in Presburger arithmetic (PA) [36]. This is the reason why model-checking problems with temporal logics have been considered for some of these classes like one-counter automata [18,19], Petri nets [22], reversal-bounded counter systems [3], flat counter systems [16]. The previous list is certainly not exhaustive and a general question is how to take advantage of the decidability of the reachability problem to conclude the decidability of model-checking problems with temporal logics. This provides a rich variety of decision problems, depending on the subclasses of counter systems and on the temporal logics at hand. If one considers the specific class of flat (a.k.a weak) counter systems (counter systems with all loops disjoint from each other), it has been shown that their reachability sets are definable in PA, see e.g. [4,7,16,5], and the model-checking of these systems with very expressive branching-time temporal logics is decidable [12]. However in this

* Work partially supported by the EU Seventh Framework Programme under grant agreement No. PIOF-GA-2011-301166 (DATAVERIF).

latter work, no precise complexity analysis on the model-checking problem is performed. As an answer to that, recently in [10] (see also [13]) a complete complexity analysis for model-checking flat counter systems with linear-time properties has been provided. In the present paper, we decide to focus on the model-checking problem considering branching-time temporal logics, more specifically we consider a variant of CTL* [14] extended with atomic propositions which allows to describe for each position in the runs counter values.

Our motivations. As we have seen, reachability problems and the verification of linear-time properties for flat counter systems have been well-studied (see e.g. [10,5]) and our goal is to understand the computational complexity of model-checking these systems with branching-time temporal logics such as CTL or CTL* (see e.g. [14]). There exist some classes of counter systems for which considering branching-time extensions, instead of simple (repeated) reachability questions, lead to undecidability. This is the case of Petri nets for which the model-checking of CTL is undecidable (with propositional variables only) whereas the reachability problem and model-checking problems with several LTL variants are known to be decidable [33,26,22]. As far as flat counter systems are concerned, we already know that we are on the safe side in the sense that decidability of model-checking with CTL* formulae is established in [12] thanks to a translation into PA. However no lower bound is provided and the translation produces a formula which is exponentially bigger than the system given in input. This is rather unsatisfactory. Our main motivation is therefore to understand the complexity of model-checking flat counter systems with branching-time logics and to design optimal algorithms.

Our contribution. We prove that the model-checking problem of flat counter systems with a version of CTL* equipped with arithmetical constraints on counter values is equivalent to the satisfiability problem for PA, modulo logarithmic-space reductions.

- For the complexity lower bound, we show that the satisfiability problem for PA can be reduced to the model-checking problem but there is no need for arithmetical constraints and for temporal operators other than `EF`.
- For the complexity upper bound, we reduce the model-checking problem to the satisfiability problem in PA by using the fact that runs in flat counter systems can be encoded by tuples of natural numbers and then the semantics for CTL* can be internalised in PA. This last fact has been already observed in [12] but herein, we provide a logarithmic-space reduction which makes a substantial difference with [12]. Indeed, we are also able to quantify over path schemas (symbolic representation of potential infinite sets of runs), but concisely. Once more, this witnesses that verification problems can be encoded efficiently into PA, see e.g. [37,17,6].
- As a consequence, we are able to get the equivalence with PA to known branching-time temporal logics stronger than $\mathrm{CTL}_{\mathtt{EF}}$ (such as CTL) and our proof works as well when considering past-time operators..

Moreover, we compare the translation with PA fragments to obtain a more precise complexity of subproblems of the model-checking problem. From the known complexity characterization of PA fragments [20], we also deduce a finer characterisation of the complexity of model-checking problem for fragments of CTL* over flat counter systems. As far as proofs are concerned, for the lower bound, we take advantage of the observation that a quantification in PA over a variable $\mathtt{z}$ can be simulated by a loop that increments a corresponding counter and there is a correspondence between the first-order quantifier $\exists$ [resp. $\forall$] and the temporal connective $\mathtt{EF}$ [resp. $\mathtt{AG}$]. For the upper bound, quantification over path schemas is done directly followed by a quantification over the number of times loops are visited. However, we provide a new way to encode runs in flat counter systems, which is rewarding complexity-wise. Not only the complexity bound we get is better than the one given in [12] but also our reduction into PA is much simpler, and therefore this leaves some hope to incorporate it in a tool which will use then some solvers for PA or fragments, see e.g. [8,30,1].

Plan of the paper. In Section 3, we provide the preliminary definitions whereas Section 4 establishes the lower-bound results thanks to a reduction of the satisfiability problem for PA into model-checking flat counter systems with $\mathrm{CTL}_{\mathtt{EF}}$. Section 5 presents the reduction of model-checking CTL* over flat counter systems to the satisfiability problem for PA. More precisely, Section 5.1 presents a way to encode runs using a small amount of integer vectors, whereas Section 5.2 shows a way to construct a PA formula to check the validity of a run represented by such integer vectors. Section 5.3 deals with the encoding of an instance of the model-checking problem for CTL* by some Presburger arithmetic formula, which leads us to the optimal complexity upper bound. Section 6 shows that the global model-checking problem can be also solved thanks to the established translation into PA. Section 7 provides a more involved translation of model-checking problem to PA in order to get a better comparison between fragments of CTL* and known PA fragments.

The present paper is an extended and completed version of [11]. It includes all the proofs and a bit more explanations in several places. Furthermore, we have added a few more results, namely the use of past-time operators in the logic and the result about global model-checking obtained as a by-product and study of the model-checking problem for the fragments of CTL*. Moreover, the material in Section 7 is new.

Omitted proofs can be found in the technical appendix.

2 Related Work

In this section, we provide a more systematic presentation of related works, maybe at the price of little redundancy with the previous section.

Model-checking problems & PA: It is well-known that many verification problems can be shown decidable by translation into decision problems on Presburger arithmetic, see e.g. [37,17,6]. A typical situation arises when the sets of reachable configurations can be defined by formulae in Presburger arithmetic, as it is done, for instance, for one-counter automata [21], reversal-bounded counter systems [25], flat counter systems [7]. These results have paved the way to further interests in model-checking of logical formulas over these subclasses.

Similar to the case of reachability sets, many different model-checking problems over the subclasses of counter systems have been studied in connection with PA. For example, in [28], the authors have translated the problem of model-checking flat freeze LTL formulas over one-counter automata to PA formulas and thereby reducing the problem to the satisfiability problem of PA. Similarly, translations of model-checking problems over reversal-bounded counter systems have been performed in [23,3].

Verification of flat counter systems: Similar to other classes of systems, PA has been used significantly for model-checking problems over flat counter systems. It was shown initially that the set of reachable configurations for flat counter systems are Presburger definable [7] (see also [5]). In [12], the authors considered a subclass of counter systems (which are flat but more powerful than those considered in the present paper), a more expressive logical specification than considered in this paper and showed a translation from the model-checking problem to PA formulas.

Model-checking problem with various logics over flat counter systems and their optimal complexity characterization have been studied. In [9], it was shown that model-checking LTL with past over flat counter systems is NP-complete. Subsequently, in [10], a PSPACE-complete characterization was shown for regular properties. A logical step in this connection as suggested in the future work section of [10], is to investigate the model-checking problem over flat counter systems in the context of branching-time temporal logics. In terms of the techniques used in this paper, it is more related to the work done in [12]. Even though the translation presented in [12] shows the decidability of the model-checking problem, the translation itself witnesses an exponential blow-up. We show that the translation was non-optimal by giving a better translation in the current paper.

3 Counter systems and branching-time temporal logics

3.1 Presburger arithmetic

Presburger arithmetic (PA) is the first-order theory of natural numbers with addition. It was introduced by M. Presburger who has shown the decidability of its satisfiability problem by quantifier elimination [36]. We recall here the syntax and interpretation of this theory.

In the paper, we consider $\mathrm{VAR} = \{\mathsf{z}_1, \mathsf{z}_2, \mathsf{z}_3, \ldots\}$, a countably infinite set of *variables*. *Terms* are expressions of the form

$$t ::= a \in \mathbb{N} \ \mid \ a.\mathsf{z} \ \mid \ t_1 + t_2,$$

where a and k are in $\mathbb{N}$ and, z belongs to VAR. A *valuation* f is a map VAR $\to \mathbb{N}$ and it can be extended to the set of all terms inductively as follows:

$$\begin{aligned} f(a) &\stackrel{\text{def}}{=} a \\ f(a\mathsf{z}) &\stackrel{\text{def}}{=} a * f(\mathsf{z}) \\ f(t_1 + t_2) &\stackrel{\text{def}}{=} f(t_1) + f(t_2). \end{aligned}$$

Given a valuation f, a variable z and a natural n, the notation $f[\mathsf{z} \mapsto n]$ represents the valuation f' such that $f'(\mathsf{z}') = f(\mathsf{z}')$ for all $\mathsf{z}' \in \text{VAR} \setminus \{\mathsf{z}\}$ and $f'(\mathsf{z}) = n$. *Formulae* of PA are defined by the grammar

$$\phi ::= \ t \leq t' \mid \ \neg\phi \ \mid \ \phi \wedge \phi \ \mid \ \exists\, \mathsf{z}\ \phi,$$

where t and t' are terms and $\mathsf{z} \in \text{VAR}$. We denote the class of formulae without any quantifier as the *linear fragment* of PA. Thus, A formula ϕ is in the *linear fragment* $\stackrel{\text{def}}{\Leftrightarrow}$ ϕ is a Boolean combination of atomic formulae of the form $t \leq t'$. The semantics for formulae in PA is defined with the satisfaction relation $\models$ which says whether a valuation f satisfies a formula ϕ: for instance $f \models t \leq t'$ $\stackrel{\text{def}}{\Leftrightarrow}$ $f(t) \leq f(t')$ and $f \models \exists\, \mathsf{z}\ \phi$ $\stackrel{\text{def}}{\Leftrightarrow}$ there is $n \in \mathbb{N}$ such that $f[\mathsf{z} \mapsto n] \models \phi$.

We write $\phi(\mathsf{z}_1, \ldots, \mathsf{z}_n)$ to denote a formula ϕ such that its free variables are among $\mathsf{z}_1, \ldots, \mathsf{z}_n$. A formula $\phi(\mathsf{z}_1, \ldots, \mathsf{z}_n)$ with $n \geq 1$, defines a set of n-tuples

$$[\![\phi(\mathsf{z}_1, \ldots, \mathsf{z}_n)]\!] \stackrel{\text{def}}{=} \{\langle f(\mathsf{z}_1), \ldots, f(\mathsf{z}_n)\rangle \in \mathbb{N}^n : f \models \phi\}.$$

For $\mathsf{v} \in \mathbb{N}^n$, we also write $\mathsf{v} \models \phi$ instead of $\mathsf{v} \in [\![\phi(\mathsf{z}_1, \ldots, \mathsf{z}_n)]\!]$. For a given PA formula ϕ, the set of free variables of ϕ is denoted by $\mathtt{free}(\phi)$. The *satisfiability problem* for PA (also written SAT(PA)) is then the decision problem that takes as input a formula ϕ and asks whether there is a valuation f such that $f \models \phi$. If such a valuation exists, we say that ϕ is *satisfiable*. Decidability of Presburger arithmetic has been shown in [36]. An exact complexity characterization is provided in [2] and a precise study with respect to the quantifier alternation is done in [20].

3.2 Counter systems

Let $\mathtt{C} = \{\mathsf{x}_1, \mathsf{x}_2, \ldots\}$ be a countably infinite set of *counters* and we write $\mathtt{C}_n$ to denote the subset $\{\mathsf{x}_1, \ldots, \mathsf{x}_n\}$. Let $\text{AT} = \{\mathsf{p}_1, \mathsf{p}_2, \ldots\}$ be a countably infinite set of propositional variables.

Definition 1. *A* counter system *is a tuple* $\langle Q, \mathtt{C}_n, \Delta, \ell\rangle$ *where*

- Q *is a finite set of* control states*,*
- $\Delta \subseteq Q \times G_n \times \mathbb{Z}^n \times Q$ *is a finite set of* transitions*, labelled by* guards *and* updates *where* G_n *is a set of linear Presburger formulae* ϕ *with* $\mathtt{free}(\phi) \subseteq \{\mathsf{x}_1, \ldots, \mathsf{x}_n\}$.
- $\ell : Q \to 2^{\text{AT}}$ *is a* labeling function*,*

Guards can be made quite general and any Presburger formula can be allowed. Since we provide a translation into PA, this can allow us to consider a more general model, as in Presburger counter machines [12,29]. However, since we will be talking about precise characterization of problems where quantifier alternation is crucial, we restrict our guards to be from the linear fragment of Presburger arithmetic

For each transition $\delta = \langle q, \mathtt{g}, \mathtt{u}, q' \rangle$ in Δ, we use the following notations: $\mathit{source}(\delta) = q$, $\mathit{target}(\delta) = q'$, $\mathit{guard}(\delta) = \mathtt{g}$ and $\mathit{update}(\delta) = \mathtt{u}$. As usual, to a counter system $S = \langle Q, \mathtt{C}_n, \Delta, \ell \rangle$, we associate a labelled transition system $\mathfrak{T}(S) = \langle \mathcal{C}, \rightarrow \rangle$ where $\mathcal{C} = Q \times \mathbb{N}^n$ is the set of *configurations* and $\rightarrow \subseteq \mathcal{C} \times \Delta \times \mathcal{C}$ is the *transition relation* defined by: $\langle \langle q, \mathtt{v} \rangle, \delta, \langle q', \mathtt{v}' \rangle \rangle \in \rightarrow$ (also written $\langle q, \mathtt{v} \rangle \xrightarrow{\delta} \langle q', \mathtt{v}' \rangle$) iff

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

We write $c \rightarrow c'$ iff there exists some edge δ such that $c \xrightarrow{\delta} c'$.

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

$$\rho := c_0 \xrightarrow{\delta_0} \cdots \xrightarrow{\delta_{m-1}} c_m \xrightarrow{\delta_m} \cdots$$

where $c_i \in Q \times \mathbb{N}^n$ and $\delta_i \in \Delta$, for all $i \in \mathbb{N}$. Then, let $\mathit{trans}(\rho)$ be the ω-word $\delta_0\delta_1 \ldots$ made of transitions appearing in the run ρ. For every $i \geq 0$, we define $\rho[i] = c_i$ and $\rho_{\leq i} = c_0 \xrightarrow{\delta_0} c_1 \cdots \xrightarrow{\delta_{i-1}} c_i$. We write $c \rightarrow^* c'$ iff there exists a run ρ and $i \in \mathbb{N}$ such that $\rho[0] = c$ and $\rho[i] = c'$. Note that a run in a counter system S is either finite or infinite. A run ρ is *maximal* iff either it is infinite, or it is finite and the last configuration is a deadlock (i.e., it does not have any successor configurations). For a finite maximal run ρ, we write $|\rho| = \alpha$ where $\rho[\alpha]$ is the deadlock configuration. For an infinite maximal run ρ, we have $|\rho| = \omega$.

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) [7,32]. In a flat counter system, simple cycles can be organized as a DAG where two simple cycles are in the relation whenever there is a path between a node of the first cycle and a node of the second cycle. We write `FlatCS` to denote the class of flat counter systems.

3.3 Logical specifications

We consider a logic that extends the classical CTL* [14] by adding past-time temporal operators and arithmetical constraints on counter values. By a slight abuse of notation, the logic is also called CTL*. Since this logic will be used to describe runs of a counter machine equipped with a set of counters $\mathtt{C}_n =$

$\{\mathtt{x}_1, \ldots, \mathtt{x}_n\}$, we assume that the dimension n is known when giving formulae. The CTL* formulae are defined as follows:

$$\phi ::= \mathtt{p} \mid \psi(\mathtt{x}_1, \ldots, \mathtt{x}_n) \mid \phi \wedge \phi \mid \neg\phi \mid \mathtt{X}\phi \mid \phi\mathtt{U}\phi \mid \mathtt{X}^{-1}\phi \mid \phi\mathtt{S}\phi \mid \mathtt{E}\phi$$

where $\mathtt{p} \in \mathrm{AT}$ and $\psi(\mathtt{x}_1, \ldots, \mathtt{x}_n)$ is a Presburger formula with free variables in $\mathtt{C}_n$. We write $\mathrm{CTL}_{\mathtt{EF}}$ to denote the fragment of CTL* in which the only (unary) temporal operator is $\mathtt{EF}$ ($\mathtt{EF}\phi \stackrel{\text{def}}{=} \mathtt{E}\ (\top\ \mathtt{U}\ \phi)$ and $\top \stackrel{\text{def}}{=} (\mathtt{x}_1 = \mathtt{x}_1)$). Our version of CTL* is defined as the standard version with past-time operators, see e.g. [14], except that the Kripke structures are replaced by transition systems from counter systems and at the atomic level, arithmetical constraints are allowed.

Let $S = \langle Q, \mathtt{C}_n, \Delta, \ell\rangle$ be a counter system with transition system $\mathfrak{T}(S) = \langle \mathcal{C}, \rightarrow\rangle$. Given a maximal run ρ in S and a position $i \leq |\rho|$, the satisfaction relation $\models$ for CTL* is defined as follows:

$$\begin{array}{lll}
\rho, i \models \mathtt{p} & \stackrel{\text{def}}{\Leftrightarrow} & \mathtt{p} \in \ell(q), \text{ where } \rho[i] = \langle q, \mathtt{v}\rangle \\
\rho, i \models \psi(\mathtt{x}_1, \ldots, \mathtt{x}_n) & \stackrel{\text{def}}{\Leftrightarrow} & \mathtt{v} \models \psi(\mathtt{x}_1, \ldots, \mathtt{x}_n), \text{ where } \rho[i] = \langle q, \mathtt{v}\rangle \\
\rho, i \models \mathtt{X}\psi & \stackrel{\text{def}}{\Leftrightarrow} & i+1 \leq |\rho| \text{ and } \rho, i+1 \models \psi \\
\rho, i \models \psi_1\mathtt{U}\psi_2 & \stackrel{\text{def}}{\Leftrightarrow} & \rho, j \models \psi_2 \text{ for some } i \leq j \\
 & & \quad \text{such that } j \leq |\rho| \text{ and } \rho, k \models \psi_1 \text{ for all } i \leq k < j \\
\rho, i \models \mathtt{X}^{-1}\psi & \stackrel{\text{def}}{\Leftrightarrow} & i > 0 \text{ and } \rho, i-1 \models \psi \\
\rho, i \models \psi_1\mathtt{S}\psi_2 & \stackrel{\text{def}}{\Leftrightarrow} & \rho, j \models \psi_2 \text{ for some } j \leq i \\
 & & \quad \text{such that } j \geq 0 \text{ and } \rho, k \models \psi_1 \text{ for all } j < k \leq i \\
\rho, i \models \mathtt{E}\phi & \stackrel{\text{def}}{\Leftrightarrow} & \text{there is a maximal run } \rho' \text{ s.t. } \rho'_{\leq i} = \rho_{\leq i} \text{ and } \rho', i \models \phi.
\end{array}$$

Given a CTL* formula ϕ, a counter system S and a configuration c from S, we say that $S, c \models \phi$ iff there exists a maximal run ρ in S with $\rho[0] = c$ such that $\rho, 0 \models \phi$. Note that the symbol $\models$ is overloaded but this shall not cause any confusion. A flat counter system S is called *non-blocking* if every maximal run ρ in S is infinite. Otherwise it is called a *blocking* flat counter system.

The model-checking problem for flat counter systems over CTL* is defined as follows (let us call it MC(CTL*, `FlatCS`)):

Input: A flat counter system S, a configuration c and a formula ϕ in CTL*
Output: Do we have $S, c \models \phi$?

We know that MC(CTL*, `FlatCS`) is decidable [12] but its exact complexity is not fully characterised (actually, this is the main purpose of the present paper). The restriction to LTL formulae is known to be NP-complete [9] when guards are restricted to the linear fragment. In Section 4, we show that SAT(PA) can be reduced to MC(CTL*, `FlatCS`) restricted to $\mathrm{CTL}_{\mathtt{EF}}$ without arithmetical constraints and to flat counter systems such that the guards are simple linear constraints. By contrast, model-checking flat finite Kripke structures over CTL* is Δ_2^P-complete [15,27].

3.4 From blocking to non-blocking flat counter systems

We shall see now that we can restrict our attention to non-blocking flat counter systems where all the maximal runs are infinite. The idea is that we can transform using logarithmic space an instance of the model-checking problem with any flat counter systems into one where the counter system is flat and non-blocking.

Let L be either CTL* or $\mathrm{CTL}_{\mathtt{EF}}$, $S = \langle Q, \Delta, \mathtt{C}_n, \ell \rangle$ a flat counter system, c a configuration and ϕ a formula in L. We will show how to build in logarithmic space a non-blocking flat counter system S', a configuration c' and a formula ϕ' in L such that $S, c \models \phi$ iff $S', c' \models \phi'$.

Roughly speaking, the counter system S' is obtained from S by adding a new state $q_\perp$ that is reachable whenever a deadlock configuration is reached in S and $q_\perp$ has a self-loop with no effect on counters. The formula ϕ' is obtained from ϕ by relativisation related to the control state $q_\perp$.

We construct $S' = \langle Q', \Delta', \mathtt{C}_n, \ell' \rangle$ as follows:

- $Q' \stackrel{\text{def}}{=} Q \uplus \{q_\perp\}$.
- $\ell'(q_\perp) \stackrel{\text{def}}{=} \{\mathtt{p}_\perp\}$ where $\mathtt{p}_\perp$ is a new propositional variable and for all $q \in Q$, $\ell'(q) \stackrel{\text{def}}{=} \ell(q)$.
- $\Delta' \stackrel{\text{def}}{=} \Delta \cup \{\langle q, \neg \mathtt{g}_q, \mathtt{0}, q_\perp \rangle | q \in Q\} \cup \{\langle q_\perp, \top, \mathtt{0}, q_\perp \rangle\}$ where for every state $q \in Q$, $\mathtt{g}_q = \bigvee_{source(\delta)=q} guard(\delta)$.

Observe that since S is flat, the counter system S' is flat too.

We define then a formula map $\mathtt{t}(\cdot)$ for formulas in CTL* such that $\mathtt{t}(\cdot)$ is the identity for the atomic formulae, is homomorphic for Boolean connectives, for the temporal operators $\mathtt{X}^{-1}$ and $\mathtt{S}$ and for the path quantifier $\mathtt{E}$ and, it satisfies the following simple clauses:

- $\mathtt{t}(\mathtt{X}\psi) \stackrel{\text{def}}{=} \mathtt{X}(\neg \mathtt{p}_\perp \wedge \mathtt{t}(\psi))$.
- $\mathtt{t}(\psi \mathtt{U} \psi') \stackrel{\text{def}}{=} \mathtt{t}(\psi) \mathtt{U} (\neg \mathtt{p}_\perp \wedge \mathtt{t}(\psi'))$.

When dealing with the fragment $\mathrm{CTL}_{\mathtt{EF}}$, these two clauses are replaced by $\mathtt{t}(\mathtt{EF}\psi) = \mathtt{EF}(\neg \mathtt{p}_\perp \wedge \mathtt{t}(\psi))$. Then, we have the following result whose detailed proof can be found in Appendix A.

Lemma 2. *S' is a non-blocking flat counter system, $\mathtt{t}(\phi')$ belongs to* L *and $S, c \models \phi$ iff $S', c \models \mathtt{t}(\phi)$.*

Thanks to Lemma 2, in the sequel only non-blocking flat counter systems are considered. Furthermore, since the reachability relation is definable in PA for flat counter systems [12], it is even possible to decide whether all maximal runs from a given configuration are infinite.

4 From SAT(PA) to a subproblem of MC(CTL*, `FlatCS`)

Atomic formulae in CTL* are arbitrary Presburger formulae with free variables in $\{\mathtt{x}_1, \ldots, \mathtt{x}_n\}$. Consequently, it is easy to reduce (in logarithmic space) SAT(PA)

to MC(CTL*, `FlatCS`). Clearly, this is not interesting and the generality of the guards and the logical atomic formulae in the paper is only considered because, when establishing the complexity upper bound, we can be quite liberal. Below, we show that a very restricted fragment of MC(CTL*, `FlatCS`), simply called MC^{-}(CTL*, `FlatCS`), is already as hard as the satisfiability problem for PA and our reduction is based on a simple correspondence between quantifiers in PA and the temporal operators `EF` and `AG` in CTL*. First, we define MC^{-}(CTL*, `FlatCS`) as the subproblem of MC(CTL*, `FlatCS`) with the two following restrictions:

1. atomic formulae are restricted to propositional variables (no arithmetical constraints on counter values) and the only temporal connective is `EF` (and its dual `AG`, by closure under negation);
2. the guards on the transitions are linear constraints $t \leq t'$ or their negations. So, there are no Boolean connectives and no first-order quantifications.

Theorem 3. *There is a logspace reduction from SAT(PA) to MC^{-}(*CTL*, `FlatCS`*).*

Proof. Let ϕ be a formula in PA. Without any loss of generality, we can assume that ϕ has the following form:

$$\mathcal{Q}_1\ \mathsf{z}_1\ \mathcal{Q}_2\ \mathsf{z}_2\ \cdots \mathcal{Q}_n\ \mathsf{z}_n\ \phi'(\mathsf{z}_1, \mathsf{z}_2, \ldots, \mathsf{z}_n)$$

with $\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_n \in \{\exists, \forall\}$ and ϕ' is a quantifier-free formula. Note that given any formula in PA, we can reduce it to an equivalent formula of this form in logarithmic space (which is then fine for our main result since logarithmic-space reductions are closed under composition [35]). This is essentially based on the construction of formulae in prenex normal form in first-order logic.

Let us consider the counter system S_ϕ defined below in Figure 1 where $\mathsf{e}_i \in \mathbb{N}^n$ is the ith unit vector. Observe that $\phi'(\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_n)$ may contain

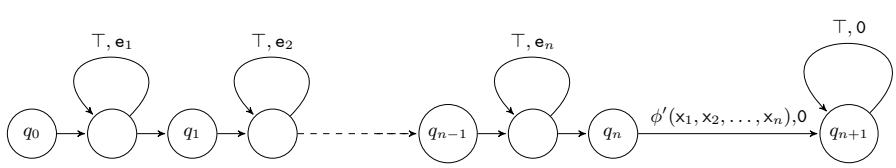


Fig. 1. Flat counter system S_ϕ

Boolean connectives but we can get rid of them in S_ϕ. In order to eliminate the Boolean connectives in the guard of the transition between q_n and q_{n+1}, we follow two simple rules, while preserving flatness (easy to check since that transition does not belong to a loop). We explain now our technique. Without any loss of generality, we can assume that negations are only in front of linear constraints. For each transition of the form $q \xrightarrow{\psi, 0} q'$ with ψ being a formula with Boolean connectives, we disintegrate it into a flat system such that there exists a run from

q to q' iff $\mathtt{v} \models \psi$ holds for any configuration $\langle q, \mathtt{v}\rangle$. A transition $q \xrightarrow{\psi_1 \wedge \psi_2, \mathtt{0}} q'$ is replaced by $q \xrightarrow{\psi_1, \mathtt{0}} q'' \xrightarrow{\psi_2, \mathtt{0}} q'$ where q'' is new. Similarly, a transition $q \xrightarrow{\psi_1 \vee \psi_2, \mathtt{0}} q'$ is replaced by $q \xrightarrow{\psi_1, \mathtt{0}} q'$ and $q \xrightarrow{\psi_2, \mathtt{0}} q'$, assuming that q does not belong to a loop. It is easy to show that S_ϕ can be transformed into a flat counter system S'_ϕ by applying the two rules above as much as possible so that eventually, S'_ϕ is a counter system matching the requirements of MC$^-$(CTL*, `FlatCS`). This is a standard way of decomposing and representing a positive Boolean combination of literals. For the ease of the proof, we will actually prove the properties of S_ϕ as shown in Figure 1.

Below, we define ψ in CTL$_{\mathtt{EF}}$ whose atomic formulae are among $q_1, \ldots, q_{n+1}$ (also abusively understood as control states) such that

(†) $S_\phi, \langle q_0, \mathtt{0}\rangle \models \psi$ iff ϕ is satisfiable in PA.

Intuitively, each variable $\mathtt{z}_i$ from ϕ is taken care of by the ith loop (that can only increment the ith counter). This is not enough, and additionally, the quantifications from ϕ are simulated in the formula ψ by using `EF` or `AG`, depending whether the first-order quantification is either existential or universal. Below, we define formally the formulae ψ_i with $i \in [1, n+1]$ so that $\psi \stackrel{\text{def}}{=} \psi_1$:

$$\psi_i \stackrel{\text{def}}{=} \begin{cases} \mathtt{EF}(q_i \wedge \psi_{i+1}) & i \leq n \text{ and } \mathcal{Q}_i = \exists \\ \mathtt{AG}(q_i \Rightarrow \psi_{i+1}) & i \leq n \text{ and } \mathcal{Q}_i = \forall \\ \mathtt{EF} q_{n+1} & i = n+1 \end{cases}$$

Given a valuation $f : \mathrm{VAR} \to \mathbb{N}$, we write $\mathtt{v}_f \in \mathbb{N}^n$ to denote the vector such that $\mathtt{v}_f[i] \stackrel{\text{def}}{=} f(\mathtt{z}_i)$ for every $i \in [1, n]$. In order to establish (†) it is sufficient to show the following property (††).

(††).1 for all valuations f, we have $f \models \phi'(\mathtt{z}_1, \mathtt{z}_2, \ldots, \mathtt{z}_n)$ iff $\langle q_n, \mathtt{v}_f\rangle \models \psi_{n+1}$,
(††).2 for all $i \in [1, n]$ and valuations f such that $f(\mathtt{z}_i) = \cdots = f(\mathtt{z}_n) = 0$, we have $f \models \mathcal{Q}\ \mathtt{z}_i\ \cdots\ \mathcal{Q}_n\ \mathtt{z}_n\ \phi'(\mathtt{z}_1, \mathtt{z}_2, \ldots, \mathtt{z}_n)$ iff $\langle q_{i-1}, \mathtt{v}_f\rangle \models \psi_i$.

It is easy to see that, we get the property (†) by applying (††).2 with $i = 1$.

First, we prove (††).1. Recall that $\psi_{n+1} = \mathtt{EF}\ q_{n+1}$ and the only edge between q_n and q_{n+1} is the following:

$$q_n \xrightarrow{\phi'(\mathtt{x}_1, \ldots, \mathtt{x}_n), \mathtt{0}} q_{n+1}$$

Let f be a valuation. By the definition of $\models$, we have $\langle q_n, \mathtt{v}_f\rangle \models \mathtt{EF}\ q_{n+1}$ iff $\mathtt{v}_f \models \phi'(\mathtt{x}_1, \ldots, \mathtt{x}_n)$. By definition of $\mathtt{v}_f$, we get $\mathtt{v}_f \models \phi'(\mathtt{x}_1, \ldots, \mathtt{x}_n)$ iff $f \models \phi'(\mathtt{z}_1, \mathtt{z}_2, \ldots, \mathtt{z}_n)$.

Now we prove (††).2 by induction. The base case $i = n+1$ corresponds precisely to the satisfaction of the property (††).1. Suppose that (††).2 holds for $i+1$, we show that it holds for i too.

Case 1: $\mathcal{Q}_i = \exists$.
Let f be a valuation such that $f(\mathtt{z}_i) = \cdots = f(\mathtt{z}_n) = 0$. We prove that $f \models \exists\ \mathtt{z}_i\ \cdots\ \mathcal{Q}_n\ \mathtt{z}_n\ \phi'(\mathtt{z}_1, \ldots, \mathtt{z}_n)$ iff $\langle q_{i-1}, \mathtt{v}_f\rangle \models \mathtt{EF}(q_i \wedge \psi_{i+1})$.

First, assume that $f \models \exists\ \mathtt{z}_i\ \cdots\ \mathcal{Q}_n\ \mathtt{z}_n\ \phi'(\mathtt{z}_1, \ldots, \mathtt{z}_n)$. By assumption, there exists $a \in \mathbb{N}$ such that $f[\mathtt{z}_i \mapsto a] \models \mathcal{Q}_{i+1}\ \mathtt{z}_{i+1}\ \cdots\ \mathcal{Q}_n\ \mathtt{z}_n\ \phi'(\mathtt{z}_1, \mathtt{z}_2, \ldots, \mathtt{z}_n)$. By the induction hypothesis, $\langle q_i, \mathtt{v}_{f'} \rangle \models \psi_{i+1}$ where $f' = f[\mathtt{z}_i \mapsto a]$. Now consider the transition system $\mathfrak{T}(S_\phi)$ and its configuration $\langle q_{i-1}, \mathtt{v}_f \rangle$. There is a run ρ in $\mathfrak{T}(S)$ starting from $\langle q_{i-1}, \mathtt{v}_f \rangle$ that reaches q_i where the loop l_i is visited exactly a times. Since the effect of the simple loop l_i is to increment the counter $\mathtt{x}_i$ each time, and $\mathtt{v}_f[i] = 0$, there is a finite run from $\langle q_{i-1}, \mathtt{v}_f \rangle$ to $\langle q_i, \mathtt{v}_{f'} \rangle$. So, $\langle q_{i-1}, \mathtt{v}_f \rangle \models \mathtt{EF}(q_i \wedge \psi_{i+1})$.

On the other hand, assume that $\langle q_{i-1}, \mathtt{v}_f \rangle \models \mathtt{EF}(q_i \wedge \psi_{i+1})$. By definition of $\models$, there exists a run ρ in $\mathfrak{T}(S_\phi)$ with $\rho[0] = \langle q_{i-1}, \mathtt{v}_f \rangle$ such that for some position $j \leq |\rho|$, $\rho[j] \models q_i \wedge \psi_{i+1}$. Thus, $\rho[j] = \langle q_i, \mathtt{v}' \rangle$ for some $\mathtt{v}' \in \mathbb{N}^n$. Since q_i is reached only after taking the simple loop l_i, let us assume that the loop l_i is taken $a \geq 0$ times. Since, the effect of the simple loop l_i is to increment the counter $\mathtt{x}_i$ each time, $\mathtt{v}'[i] = a$ and $\mathtt{v}_f$ and $\mathtt{v}'$ differ only on the ith component. So, $\mathtt{v}' = \mathtt{v}_{f'}$ where $f' = f[\mathtt{z}_i \mapsto a]$. Since $\langle q_i, \mathtt{v}_{f'} \rangle \models \psi_{i+1}$, by the induction hypothesis $f[\mathtt{z}_i \mapsto a] \models \mathcal{Q}_{i+1}\ \mathtt{z}_{i+1}\ \cdots\ \mathcal{Q}_n\ \mathtt{z}_n\ \phi'(\mathtt{z}_1, \mathtt{z}_2, \ldots, \mathtt{z}_n)$. By definition of $\models$ in PA, we conclude that $f \models \exists\ \mathtt{z}_i\ \cdots\ \mathcal{Q}_n\ \mathtt{z}_n\ \phi'(\mathtt{z}_1, \ldots, \mathtt{z}_n)$.

Case 2: $\mathcal{Q}_i = \forall$.
Let f be a valuation such that $f(\mathtt{z}_i) = \cdots = f(\mathtt{z}_n) = 0$. We prove that $f \models \forall\ \mathtt{z}_i\ \cdots\ \mathcal{Q}_n\ \mathtt{z}_n\ \phi'(\mathtt{z}_1, \ldots, \mathtt{z}_n)$ iff $\langle q_{i-1}, \mathtt{v}_f \rangle \models \mathtt{AG}(q_i \Rightarrow \psi_{i+1})$.

First, we assume that $f \models \forall\ \mathtt{z}_i\ \cdots\ \mathcal{Q}_n\ \mathtt{z}_n\ \phi'(\mathtt{z}_1, \ldots, \mathtt{z}_n)$. By assumption, for all $a \in \mathbb{N}$ we have $f[\mathtt{z}_i \mapsto a] \models \mathcal{Q}_{i+1}\ \mathtt{z}_{i+1}\ \cdots\ \mathcal{Q}_n\ \mathtt{z}_n\ \phi'(\mathtt{z}_1, \mathtt{z}_2, \ldots, \mathtt{z}_n)$. By the induction hypothesis, we have $\langle q_i, \mathtt{v}_{f'} \rangle \models \psi_{i+1}$ where $f' = f[\mathtt{z}_i \mapsto a]$ with $a \in \mathbb{N}$. Now consider the transition system $\mathfrak{T}(S_\phi)$ and its configuration $\langle q_{i-1}, \mathtt{v}_f \rangle$. There are two types of run ρ in $\mathfrak{T}(S)$ starting from $\langle q_{i-1}, \mathtt{v}_f \rangle$:

1. ρ reaches q_i and the loop l_i is visited exactly a times for some $a \in \mathbb{N}$.
2. ρ loops l_i infinitely many times and never reaches q_i.

Due to the implication, $q_i \Rightarrow \psi_{i+1}$ we can easily ignore the runs of type 2 above. Hence, we will consider only the runs where l_i is taken only finitely many times. Since the effect of the simple loop l_i is to increment the counter $\mathtt{x}_i$ each time, and $\mathtt{v}_f[i] = 0$, there are finite runs from $\langle q_{i-1}, \mathtt{v}_f \rangle$ to $\langle q_i, \mathtt{v}_{f'} \rangle$ for each of the valuations $f' = f[\mathtt{z}_i \mapsto a]$ with $a \in \mathbb{N}$. So, $\langle q_{i-1}, \mathtt{v}_f \rangle \models \mathtt{AG}(q_i \Rightarrow \psi_{i+1})$.

Now, assume that $\langle q_{i-1}, \mathtt{v}_f \rangle \models \mathtt{AG}(q_i \Rightarrow \psi_{i+1})$. By definition of $\models$, there exists a set of runs X, in $\mathfrak{T}(S_\phi)$ such that for all $\rho \in X$ we have $\rho[0] = \langle q_{i-1}, \mathtt{v}_f \rangle$ and for all positions $j < |\rho|$, if $\rho[j] \models q_i$ then $\rho[j] \models \psi_{i+1}$. Thus, for every run $\rho \in X$, $\rho[j] = \langle q_i, \mathtt{v}' \rangle$ for some $\mathtt{v}' \in \mathbb{N}^n$. Since q_i is reached only after taking the simple loop l_i a finite number of times in any run, let us assume that for each value of $a \geq 0$, there exists a run $\rho_a \in X$ where the loop l_i is taken a times. Since the effect of the simple loop l_i is to increment the counter $\mathtt{x}_i$ each time, we have $\mathtt{v}'[i] = a$ and $\mathtt{v}_f$ and $\mathtt{v}'$ differ only on the ith component in any run ρ_a in X. So, $\mathtt{v}' = \mathtt{v}_{f'}$ where $f' = f[\mathtt{z}_i \mapsto a]$. Since $\langle q_i, \mathtt{v}_{f'} \rangle \models \psi_{i+1}$ for all such valuations $\mathtt{v}_{f'}$, by the induction hypothesis $f[\mathtt{z}_i \mapsto a] \models \mathcal{Q}_{i+1}\ \mathtt{z}_{i+1}\ \cdots\ \mathcal{Q}_n\ \mathtt{z}_n\ \phi'(\mathtt{z}_1, \mathtt{z}_2, \ldots, \mathtt{z}_n)$ for all $a \in \mathbb{N}$. By definition of $\models$ in PA, we conclude that $f \models \forall\ \mathtt{z}_i\ \cdots\ \mathcal{Q}_n\ \mathtt{z}_n\ \phi'(\mathtt{z}_1, \ldots, \mathtt{z}_n)$. $\square$

Thus, there is a logarithmic-space reduction from SAT(PA) into $\mathrm{MC}^-(\mathrm{CTL}^*$, `FlatCS`). This gives one side of the complexity equivalence we are currently establishing.

5 From MC(CTL*, `FlatCS`) to SAT(PA)

In this section, we present a logarithmic-space reduction from MC(CTL*, `FlatCS`) to the satisfiability problem for PA. In [12], a reduction is already presented to get decidability of MC(CTL*, `FlatCS`). Unfortunately, it requires exponential space and it is quite difficult to parse. Following a similar idea, we propose here a simpler reduction that has the great advantage to be optimal complexity-wise. The idea of this reduction is based on the two following points:

1. encoding the runs in flat counter systems by tuples of natural numbers thanks to a symbolic representation for potential infinite sets of runs, see path schemas in [9];
2. internalising CTL* semantics into (PA) by using the encoding of runs.

Below, we consider a fixed non-blocking flat counter system $S = \langle Q, \mathtt{C}_n, \Delta, \ell \rangle$. Without any loss of generality we can assume that $Q = \{1, \ldots, \alpha\}$ for some $\alpha \geq 1$ and $\Delta = \{\delta_1, \ldots, \delta_\beta\}$ for some $\beta \geq 1$. Since $Q \subseteq \mathbb{N}$, configurations of S can be seen as vectors in $\mathbb{N}^{n+1}$ where the first component represents the control state and range within the interval $[1, \alpha]$ and the rest represents counter values of the configuration.

5.1 Minimal path schemas

In [9], following an idea from [31], *minimal path schemas* are introduced as a means to symbolically represent all runs in flat counter systems. Path schemas can be defined as finite sequences made of transitions or simple loops (conditions apply). Formal definition is recalled below. A *simple loop* l of S is a non-empty finite sequence of transitions $\delta_1, \ldots, \delta_m$ such that $\mathit{source}(\delta_1) = \mathit{target}(\delta_m)$, $\mathit{source}(\delta_j) = \mathit{target}(\delta_{j+1})$ for all $j \in [1, m-1]$, and, for all $j, k \in [1, m]$, if $j \neq k$ then $\delta_j \neq \delta_k$ (no repetition). The length of l, written $\mathit{length}(l)$, is the number m and we denote by $\mathit{source}(l) = \mathit{target}(l)$ the control state $\mathit{source}(\delta_1)$. The number of simple loops is necessarily finite and we assume that the set of loops of S is $L = \{l_1, l_2, \ldots, l_\gamma\}$. Since S is flat, we have $\gamma \leq \alpha$. Note that since there is no restriction on path schema, there can be redundant loops and edges. To restrict this, we define a minimal path schema with no redundancy, as follows. A *minimal path schema* P is a non-empty sequence $u_1, \ldots, u_N$ such that each $u_i \in \Delta \cup L$ and the following conditions are satisfied.

1. u_N is a loop,
2. $i \neq j$ implies $u_i \neq u_j$,
3. for all $i \in [1, N-1]$, we have $\mathit{target}(u_i) = \mathit{source}(u_{i+1})$.

The second condition guarantees that there are no redundancies whereas the third condition ensures that P respects the control graph of S. The *size* of P, denoted by $size(P)$, is equal to N. For all $j \in [1, N]$, we write $P[j]$ for u_j. Here is an obvious result.

Lemma 4. *The size of a minimal path schema is bounded by* $\beta + \gamma \leq \beta + \alpha$.

In order to obtain concrete paths from a path schema P, we augment P with a vector specifying how many times each internal loop is visited. By definition, a loop in P is *internal* if it is not the last one. An *iterated path schema* is a pair $\langle P, \mathtt{m} \rangle$ where P is a minimal path schema and $\mathtt{m} \in \mathbb{N}^{size(P)}$ such that $\mathtt{m}[1] = size(P)$ and for all $i \in [1, size(P) - 1]$, $\mathtt{m}[i+1] > 0$ and if $P[i] \in \Delta$, then $\mathtt{m}[i+1] = 1$. From $\langle P, \mathtt{m} \rangle$, we define the ω-word

$$trans(P, \mathtt{m}) \stackrel{\text{def}}{=} P[1]^{\mathtt{m}[2]} \dots P[j]^{\mathtt{m}[j+1]} \dots P[size(P) - 1]^{\mathtt{m}[size(P)]} P[size(P)]^{\omega}$$

Note that, by definition of minimal path schema, $P[size(P)]$ is a loop. Lemma 5 below states that iterated path schemas encode all the runs in flat counter systems thanks to flatness.

Lemma 5. *[9] Given an infinite run ρ in a (non-blocking) flat counter system S, there exists an iterated path schema $\langle P, \mathtt{m} \rangle$ such that $trans(\rho) = trans(P, \mathtt{m})$.*

Encoding iterated path schemas. Thanks to Lemma 4, we can encode path schemas by vectors in $\mathbb{N}^K$ with $K = 1 + \beta + \gamma$. Intuitively, we encode a path schema P by two vectors $\mathtt{v}_p$ and $\mathtt{v}_t$ in $\mathbb{N}^K$ where the first element of each vector is equal to $size(P)$ (by convention) and for all $i \in [2, size(P) + 1]$, we have $\mathtt{v}_t[i] = 1$ if $P[i]$ is a loop and $\mathtt{v}_t[i] = 0$ otherwise. So, $\mathtt{v}_t$ encodes the *type* of each element (transition vs. loop) in the sequence defining P. Similarly, $\mathtt{v}_p[i]$ represents the number of the associated transition or loop; for instance, $\mathtt{v}_p[i] = 2$ and $\mathtt{v}_t[i] = 1$ encodes that $P[i]$ is the second loop, say l_2. Then, we encode the vector $\mathtt{m}$ by a vector $\mathtt{v}_{it} \in \mathbb{N}^K$. Let us formalize this. First, we define the function $\tau : (\{0\} \times [1, \beta]) \cup (\{1\} \times [1, \gamma])) \to \Delta \cup L$ such that $\tau(0, i) \stackrel{\text{def}}{=} \delta_i$ and $\tau(1, i) \stackrel{\text{def}}{=} l_i$. Now, we provide a set of conditions $\mathfrak{C}$ on the vectors $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \in \mathbb{N}^K$ so that we can build from these three vectors an iterated path schema.

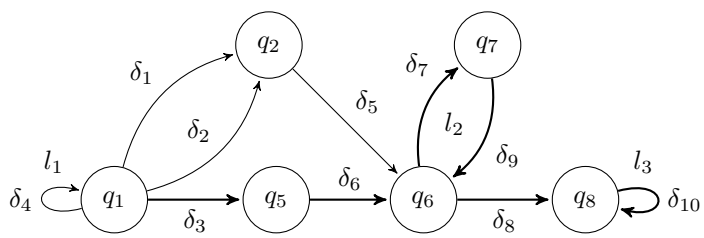
$\mathfrak{C}.1$ $\mathtt{v}_p[1] = \mathtt{v}_t[1] = \mathtt{v}_{it}[1]$ with $\mathtt{v}_p[1] \in [1, K-1]$;
for all $i \in [\mathtt{v}_{it}[1] + 2, K]$, $\mathtt{v}_p[i] = \mathtt{v}_t[i] = \mathtt{v}_{it}[i] = 0$,
$\mathfrak{C}.2$ $\mathtt{v}_t[i] \in \{0, 1\}$ for all $i \in [2, K]$,
$\mathfrak{C}.3$ if $\mathtt{v}_t[i] = 0$ then $\mathtt{v}_p[i] \in [1, \beta]$, for all $i \in [2, \mathtt{v}_p[1] + 1]$,
$\mathfrak{C}.4$ if $\mathtt{v}_t[i] = 1$ then $\mathtt{v}_p[i] \in [1, \gamma]$, for all $i \in [2, \mathtt{v}_p[1] + 1]$,
$\mathfrak{C}.5$ $\mathtt{v}_t[\mathtt{v}_p[1] + 1] = 1$,
$\mathfrak{C}.6$ there are no $i, j \in [2, \mathtt{v}_p[1] + 1]$ such that $i \neq j$, $\mathtt{v}_t[i] = \mathtt{v}_t[j]$ and $\mathtt{v}_p[i] = \mathtt{v}_p[j]$,
$\mathfrak{C}.7$ $target(\tau(\mathtt{v}_t[i], \mathtt{v}_p[i])) = source(\tau(\mathtt{v}_t[i+1], \mathtt{v}_p[i+1]))$ for all $i \in [2, \mathtt{v}_p[1]]$,
$\mathfrak{C}.8$ for all $i \in [2, \mathtt{v}_p[1]]$, $\mathtt{v}_{it}[i] > 0$ and if $\mathtt{v}_t[i] = 0$ then $\mathtt{v}_{it}[i] = 1$.

The first four conditions ensure that the vectorial representation is coherent. The three next conditions guarantee that the encoding respects the structure of

a minimal path schema, i.e. that the last element is a loop ($\mathfrak{C}$.5), that there are no two identical transitions or loops in the schema ($\mathfrak{C}$.6) and that the succession of elements effectively represents a path in the counter system ($\mathfrak{C}$.7). The last condition ensures that $\mathtt{v}_{it}$ matches the definition of the vector in an iterated path schema. It follows that given vectors $\mathtt{v}_p$, $\mathtt{v}_t$ and $\mathtt{v}_{it}$ in $\mathbb{N}^K$ that satisfy all the conditions ($\mathfrak{C}.i$), we can build a minimal path schema $P_{\mathtt{v}_t,\mathtt{v}_p}$ equal to

$$\tau(\mathtt{v}_t[2], \mathtt{v}_p[2]) \cdots \tau(\mathtt{v}_t[\mathtt{v}_p[1]+1], \mathtt{v}_p[\mathtt{v}_p[1]+1]).$$

From the vector $\mathtt{v}_{it}$, we can define the vector $\mathtt{m}_{\mathtt{v}_{it}} \in \mathbb{N}^{\mathtt{v}_{it}[1]}$ such that for all $i \in [1, \mathtt{v}_{it}[1]]$, $\mathtt{m}_{\mathtt{v}_{it}}[i] \stackrel{\text{def}}{=} \mathtt{v}_{it}[i+1]$. Let us explain the encoding of iterated path schemas with vectors on a simple example. We consider the following flat counter system.



There are 10 transitions and 3 simple loops. The enumeration of edges and loops is done as shown above. Let $\langle P, \mathtt{m}\rangle$ be such that

$$P = \delta_3 \cdot \delta_6 \cdot l_2 \cdot \delta_8 \cdot l_3$$

and $\mathtt{m} = (5, 1, 1, 146, 1)$. So, we get the resulting ω-word as $\delta_3 \cdot \delta_6 \cdot (l_2)^{146} \cdot \delta_8 \cdot (l_3)^\omega$. From the previous encoding, the ω-word is encoded by the following vectors

- $\mathtt{v}_p = (5, 3, 6, 2, 8, 3, 0, 0, 0, 0, 0, 0, 0, 0)$,
- $\mathtt{v}_t = (5, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0)$ and
- $\mathtt{v}_{it} = (5, 1, 1, 146, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$.

Lemma 6 below, whose proof can be found in Appendix B, states that there exists a Presburger formula $\mathit{Schema}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it})$ over the sets of variables $\mathsf{Z}_p = \{\mathsf{z}_p^1, \ldots, \mathsf{z}_p^K\}$, $\mathsf{Z}_t = \{\mathsf{z}_t^1, \ldots, \mathsf{z}_t^K\}$ and $\mathsf{Z}_{it} = \{\mathsf{z}_{it}^1, \ldots, \mathsf{z}_{it}^K\}$ which expresses the conjunctions of the conditions $(\mathfrak{C}.i)_{i\in[1,8]}$. For three vectors $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \in \mathbb{N}^K$, we say that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \models \mathit{Schema}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it})$ iff for some valuation function f, we have $f \models \mathit{Schema}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it})$, where $f(\mathsf{z}_t^i) = \mathtt{v}_t[i], f(\mathsf{z}_p^i) = \mathtt{v}_p[i]$ and $f(\mathsf{z}_{it}^i) = \mathtt{v}_{it}[i]$ for all $i \in [1, K]$.

Lemma 6.

1. *Let P be in $(\Delta \cup L)^+$ of length $N \leq \beta + \gamma$ and $\mathtt{m} \in \mathbb{N}^N$. Then, $\langle P, \mathtt{m}\rangle$ is an iterated path schema iff there are $\mathtt{v}_t$, $\mathtt{v}_p$ and $\mathtt{v}_{it}$ in $\mathbb{N}^K$ respecting $\mathfrak{C}$ such that $P = P_{\mathtt{v}_t,\mathtt{v}_p}$ and $\mathtt{m} = \mathtt{m}_{\mathtt{v}_{it}}$.*
2. *One can build a quantifier-free PA formula $\mathit{Schema}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it})$ of polynomial size in the size of the counter system S such that for all $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \in \mathbb{N}^K$, we have $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \models \mathit{Schema}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it})$ iff $\mathtt{v}_t$, $\mathtt{v}_p$ and $\mathtt{v}_{it}$ satisfy $\mathfrak{C}$.*

5.2 Encoding runs using vectors

So, we are able to characterise the minimal path schemas of a flat counter system S using a Presburger arithmetic formula. Let us show how to describe the runs respecting any path schema P of S. Lemma 5 states that any infinite run can be encoded by an iterated path schema. However, not every iterated path schema corresponds to a run in due form because counter values should be non-negative and transitions can be fired only if the guards are really satisfied. We will see now how to check that an iterated path schema indeed represents a run starting from a given configuration c. First, we need to introduce a notion of *pseudo-run* in which only the updates are precise and where the configurations can take their values in the integers (instead of in the set of natural numbers). Given $c \in Q \times \mathbb{Z}^n$, a *pseudo-run* ρ starting from c in S is an infinite sequence $\rho := c_0 \overset{\delta_1}{\rightsquigarrow} \cdots \overset{\delta_{m-1}}{\rightsquigarrow} c_m \overset{\delta_m}{\rightsquigarrow} \cdots$ where $c_0 = c$, $c_i = \langle q_i, \mathtt{v}_i \rangle \in Q \times \mathbb{Z}^n$ for all $i \geq 0$ and for all transitions $\delta \in \Delta$ we have $\langle q, \mathtt{v} \rangle \overset{\delta}{\rightsquigarrow} \langle q', \mathtt{v}' \rangle \overset{\text{def}}{\Leftrightarrow} q = \mathit{source}(\delta), q' = \mathit{target}(\delta)$ and $\mathtt{v}' = \mathtt{v} + \mathit{update}(\delta)$. So, a pseudo-run $\rho = \langle q_0, \mathtt{v}_0 \rangle \overset{\delta_1}{\rightsquigarrow} \cdots \overset{\delta_{m-1}}{\rightsquigarrow} \langle q_m, \mathtt{v}_m \rangle \cdots$ is a run iff for all $i \in \mathbb{N}$, $\mathtt{v}_i \models \mathit{guard}(\delta_i)$ and $\mathtt{v}_i \in \mathbb{N}^n$.

Note also that for all configurations $\langle q, \mathtt{v} \rangle$, if $\langle P, \mathtt{m} \rangle$ is an iterated path schema such that $\mathit{source}(P[1]) = q$ then there exists a pseudo-run ρ starting from c such that $\mathit{trans}(\rho) = \mathit{trans}(P, \mathtt{m})$ (where the function $\mathit{trans}(.)$ is defined for infinite pseudo-run the same way it is done for infinite runs, see Section 3.2). As with runs, we denote $\rho[i]$ the $(i+1)$-th configuration of a pseudo-run ρ. From these observations, we conclude that an iterated path schema $\langle P, \mathtt{m} \rangle$ augmented with $c_0 = \langle q_0, \mathtt{v}_0 \rangle$ satisfying $\mathit{source}(P[1]) = q_0$, defines a unique pseudo-run that is denoted by $\rho(P, \mathtt{m}, c_0)$. Given a configuration $c_0 = \langle q_0, \mathtt{v}_0 \rangle$, we say that $\rho(P, \mathtt{m}, c_0)$ is *well-defined* if $\mathit{source}(P[1]) = q_0$.

We introduce a sequence of positions in a pseudo-run as $\mathfrak{p}_0, \ldots, \mathfrak{p}_{\text{size}(P)-1}$. By convention $\mathfrak{p}_0 = 0$ and for every $i \in [1, \text{size}(P) - 1]$ let $\mathfrak{p}_i \overset{\text{def}}{=} \sum_{j=1}^{i} \mathtt{m}[j+1] * \mathit{length}(P[j])$. The value $\mathfrak{p}_i$ is called the *position* of the *ith witness configuration* $\rho(P, \mathtt{m}, c_0)[\mathfrak{p}_i]$. Intuitively, we reach the ith witness configuration after going through the iterations of first i elements of the path schema P in $\rho(P, \mathtt{m}, c_0)$. We say that $\rho(P, \mathtt{m}, c_0)$ is *positive* if for all the witness configurations $\langle q, \mathtt{v} \rangle$, we have $\mathtt{v} \in \mathbb{N}^n$. Note that since for all path schemas P, we have $\text{size}(P) \leq \beta + \gamma$, the number of witness configurations for such a pseudo-run is bounded by $\beta + \gamma$ too.

Now, we show how to build a PA formula whose set of solutions corresponds to the witness configurations of a pseudo-run associated to an iterated path schema equipped with an initial configuration. Before defining the formula, we explain some further notions. In the sequel, we use the following sets of variables

- $\mathsf{X}_0 = \{\mathsf{x}_0^1, \ldots, \mathsf{x}_0^{n+1}\}$ and $\mathsf{X} = \{\mathsf{x}^1, \ldots, \mathsf{x}^{n+1}\}$ to represent configurations,
- $\mathsf{W}_i = \{\mathsf{w}_i^1, \ldots, \mathsf{w}_i^{n+1}\}$ for every $i \in [0, \beta + \gamma - 1]$ to represent pseudo-configurations (we recall that $Q \subseteq \mathbb{N}$, hence (pseudo)-configurations are seen as tuples in $\mathbb{N}^{n+1}$) and,
- the variables $\mathtt{d}_0, \ldots, \mathtt{d}_{\beta+\gamma-1}$ and $\mathtt{y}$ to represent positions in (pseudo)-runs.

Furthermore, given sets of variables X, W representing (pseudo)-configurations, a variable x and a vector $\mathtt{u} \in \mathbb{N}^n$, we use the shortcut $\mathsf{X} = \mathsf{W} + \mathsf{x}.\mathtt{u}$ for the

following formula:

$$\bigwedge_{i=2}^{n+1} \mathsf{x}^i = \mathsf{w}^i + \mathsf{x.u}[i-1].$$

Now, we introduce the formula *Witness* which allows us to check whether a given set of configurations and natural numbers represent the witness configurations and their respective positions in a pseudo-run associated to an iterated path schema. The main idea of the formula is to verify at each step whether the control states of the witness configurations match with the states of the taken transitions or loops in the path schema and then to compute the effect of the corresponding element of the iterated path schema taking into account the number of iterations.

$$\begin{gathered}\mathit{Witness}(\mathsf{W}_0, \ldots, \mathsf{W}_{\beta+\gamma-1}, \mathsf{d}_0, \ldots, \mathsf{d}_{\beta+\gamma-1}, \mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0) \stackrel{\text{def}}{=} \\ (\mathsf{d}_0 = 0 \wedge \mathsf{X}_0 = \mathsf{W}_0 \wedge \bigvee_{s=0}^{1} \bigvee_{j=1}^{\max(\beta,\gamma)} \mathsf{z}_p^2 = j \wedge \mathsf{z}_t^2 = s \wedge \mathsf{x}_0^1 = \mathit{source}(\tau(s,j))) \wedge \\ \bigwedge_{i=1}^{\beta+\gamma-1} (i < \mathsf{z}_t^1 \Rightarrow \bigvee_{s=0}^{1} \bigvee_{j=1}^{\max(\beta,\gamma)} (\mathsf{z}_p^{i+1} = j \wedge \mathsf{z}_t^{i+1} = s \wedge \mathsf{d}_i = \mathsf{d}_{i-1} + \mathsf{z}_{it}^{i+1} * \mathit{length}(\tau(s,j)) \wedge \\ \mathsf{w}_i^1 = \mathit{target}(\tau(s,j)) \wedge \mathsf{W}_i = \mathsf{W}_{i-1} + \mathsf{z}_{it}^{i+1} * \mathit{update}(\tau(s,i)))\end{gathered}$$

We are aware that the above formula is not particularly pretty at first glance, but it only states simple properties. We need to go through this process of writing down such formulae for the sake of checking the correctness of our enterprise and for paving the way to a future implementation of the translation that requires sufficiently details. More formulae will follow for the same reasons but we shall try to explain as much as possible every bit, while trying to avoid the boredom of repetitive arguments.

Lemma 7 below characterizes the property respected by formula *Witness* by construction (a detailed proof can be found in Appendix C).

Lemma 7. *Let* $\mathtt{w}_0, \ldots, \mathtt{w}_{\beta+\gamma-1}, \mathtt{c}_0 \in \mathbb{N}^{n+1}$, $\mathfrak{p}_0, \ldots, \mathfrak{p}_{\beta+\gamma-1} \in \mathbb{N}$ *and* $\mathtt{v}_t$, $\mathtt{v}_p$, $\mathtt{v}_{it}$ *in* $\mathbb{N}^K$ *such that* $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \models \mathit{Schema}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it})$. *We have*

$$\mathtt{w}_0, \ldots, \mathtt{w}_{\beta+\gamma-1}, \mathfrak{p}_0, \ldots, \mathfrak{p}_{\beta+\gamma-1}, \mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0 \models \mathit{Witness}$$

iff $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)$ *is well-defined and positive and for all* $j \in [0, \beta+\gamma-1]$, *if* $j < \mathrm{size}(P_{\mathtt{v}_t,\mathtt{v}_p})$, *then* $\mathtt{w}_j$ *represents the* j*th witness configuration of* $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)$ *and* $\mathfrak{p}_j$ *its position.*

Hence, the formula *Witness* gives us a way to characterise the witness positions and the configuration at those positions of a pseudo-run represented by an iterated path schema. Identifying witness positions and the configurations is essential to check whether a pseudo-run is a run or not. In fact, using *Witness*, one can build in logarithmic space a formula $\mathit{Conf}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}, \mathsf{X})$ such that if $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i, \mathtt{c} \models \mathit{Conf}$ then the vector $\mathtt{c}$ is the ith configuration $\langle q_i, \mathtt{v}_i \rangle$ of a pseudo-run $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)$ with the property that $\mathtt{v}_i \models$

$guard(trans(\langle P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}\rangle)[i+1])$ and $\mathtt{v}_i \in \mathbb{N}^n$ (here i is the number of transitions to reach that configuration). This offers us the ability to construct then a formula $Run(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0)$ to check whether a pseudo-run is a run, since, as observed earlier, it is enough to check whether at each step the ith configuration satisfies the guard of the $(i+1)$th transition.

First, we explain how to build the formula $Conf(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}, \mathsf{X})$. The formula uses $Schema(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it})$ in order to check whether the variables correspond to an iterated path schema. Then, thanks to existential quantifications, the formula guesses the positions and the values of the witness configurations using the formula $Witness(\mathsf{W}_0, \ldots, \mathsf{W}_{\beta+\gamma-1}, \mathtt{d}_0, \ldots, \mathtt{d}_{\beta+\gamma-1}, \mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0)$ and Lemma 7. We recall that the variables in X represent the configuration at the position y in the corresponding pseudo-run. To do so, in the formula $Conf$, a case analysis is performed depending where y lays with respect to the positions of the witness configurations. Here are the cases.

1. The value stored in y corresponds to a witness position from which the next step in the path schema is an edge δ. It is then easy to find the value of the associated configuration which corresponds to a witness configuration and also thanks to the encoding of the path schema provided by the variables in Z_t ,Z_p and in Z_{it} to check that the guards of δ is satisfied.

2. The value stored in y corresponds to a position between two consecutive witness positions and lays inside a loop. One has to check how many times the corresponding loop has been taken and thanks to this information, the formula can compute the value of the reached configuration and find the corresponding guard to be checked for satisfiability.

3. The value stored in y lies in the last loop of the path schema, then the treatment is the same as for the second case.

Before defining the formula $Conf$, we introduce some useful notations. Periodicity constraints can be defined in Presburger arithmetic and we write $t \equiv_c t'$ ($c \in \mathbb{N}$ and t, t' are terms) to denote the formula $\exists \mathsf{x}\ (t = c.\mathsf{x} + t') \vee (t' = c.\mathsf{x} + t)$. Then for a simple loop l of the form $\delta_1, \ldots, \delta_m$, for all $i \in [1, m]$, we will use the notation $l[i]$ to represent the transitions δ_i. Furthermore, we write $update(l_k)$ to denote the tuple $update(\delta_1) + \cdots + update(\delta_m)$. Now, we are ready to define the formula $Conf$:

$$
\begin{aligned}
&\mathit{Conf}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}, \mathsf{X}) \stackrel{\text{def}}{=} \mathit{Schema}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}) \wedge \\
&\exists \mathsf{W}_0 \ldots \exists \mathsf{W}_{\beta+\gamma-1} \exists \mathsf{d}_0 \ldots \exists \mathsf{d}_{\beta+\gamma-1} \\
&\mathit{Witness}(\mathsf{W}_0, \ldots, \mathsf{W}_{\beta+\gamma-1}, \mathsf{d}_0, \ldots, \mathsf{d}_{\beta+\gamma-1}, \mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0) \wedge \\
&\bigvee_{j=0}^{\beta+\gamma-1} [(j \leq \mathsf{z}_p^1 \wedge \mathsf{y} = \mathsf{d}_j \wedge \mathsf{z}_t^{j+2} = 0 \wedge \mathsf{X} = \mathsf{W}_j \wedge \\
&\bigvee_{k=1}^{\beta} (\mathsf{z}_p^{j+2} = k \wedge \mathit{guard}(\delta_k)(\mathsf{x}^2, \cdots, \mathsf{x}^{n+1}))) \vee \\
&(j \leq \mathsf{z}_p^1 \wedge \mathsf{d}_j \leq \mathsf{y} < \mathsf{d}_{j+1} \wedge \mathsf{z}_t^{j+2} = 1 \wedge \\
&\bigvee_{k=1}^{\gamma} (\mathsf{z}_p^{j+2} = k \wedge \mathit{ConfLoop}_{l_k}(\mathsf{W}_j, \mathsf{y} - \mathsf{d}_j, \mathsf{X}))) \vee \\
&(j = \mathsf{z}_p^1 \wedge \mathsf{d}_j \leq \mathsf{y} \wedge \bigvee_{k=1}^{\gamma} (\mathsf{z}_p^{j+2} = k \wedge \mathit{ConfLoop}_{l_k}(\mathsf{W}_j, \mathsf{y} - \mathsf{d}_j, \mathsf{X})))]
\end{aligned}
$$

where for all loop l_k with $k \in [1, \gamma]$, the formula $\mathit{ConfLoop}_{l_k}(\mathsf{W}, \mathsf{y}, \mathsf{X})$ is used to find the configurations corresponding to X obtained when taking the loop l_k from the configuration W and taking y transitions in the loop.

$$
\begin{aligned}
&\mathit{ConfLoop}_{l_k}(\mathsf{W}, \mathsf{y}, \mathsf{X}) \stackrel{\text{def}}{=} \\
&\bigvee_{j=0}^{\mathit{length}(l_k)-1} [\mathsf{y} \equiv_{\mathit{length}(l_k)} j \wedge \mathsf{x}^1 = \mathit{source}(l_k[j+1]) \wedge \\
&\mathit{guard}(l_k[j+1])(\mathsf{x}^2, \cdots, \mathsf{x}^{n+1}) \wedge \\
&\exists \mathsf{h}.(\mathsf{h} * \mathit{length}(l_k) \leq \mathsf{y} < \mathsf{h} * \mathit{length}(l_k) + 1 \wedge \\
&\mathsf{X} = \mathsf{W} + \mathsf{h} * \mathit{update}(l_k) + \sum_{r=1}^{r=j} \mathit{update}(l_k[r]))].
\end{aligned}
$$

In the definition of the formula *Conf*, the fourth and fifth lines correspond to the first case we have stated above, the fifth and sixth lines to the second case and the last line to the last and third cases.

Furthermore, as we have already stated, a pseudo-run $\rho = \langle q_0, \mathsf{v}_0 \rangle \stackrel{\delta_1}{\rightsquigarrow} \cdots \stackrel{\delta_{m-1}}{\rightsquigarrow} \langle q_m, \mathsf{v}_m \rangle \cdots$ is a run iff for all $i \in \mathbb{N}$, $\mathsf{v}_i \models \mathit{guard}(\delta_i)$ and $\mathsf{v}_i \in \mathbb{N}^n$. Consequently to construct the formula $\mathit{Run}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0)$, we only have to check that the values contained in the set of variables Z_t, Z_p,Z_{it} and X_0 correspond to a pseudo-run for which at each step the vector value of the configuration is in $\mathbb{N}^n$ and it satisfies the guard of the next transition. According to the properties satisfied by the formula *Conf*, we build the formula *Run* as follows:

$$
\mathit{Run}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0) \stackrel{\text{def}}{=} \forall\ \mathsf{y}\ \exists\ \mathsf{X}\ \mathit{Conf}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}, \mathsf{X}).
$$

By construction we can then deduce the following Lemma (whose proof can be found in Appendix D) concerning the properties of the formulae *Conf* and *Run*

Lemma 8. *For all* $\mathsf{c}_0, \mathsf{c} \in \mathbb{N}^{n+1}$*, for all* $i \in \mathbb{N}$ *and for all* $\mathsf{v}_t, \mathsf{v}_p, \mathsf{v}_{it} \in \mathbb{N}^K$*, we have:*

1. $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i, \mathtt{c} \models$ *Conf iff* $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \models$ *Schema,* $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)$ *is well-defined,* $\mathtt{c} = \rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, c_0)[i]$ *and*

$$\mathtt{c}[2], \cdots, \mathtt{c}[n+1] \models \mathit{guard}(\mathit{trans}(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}})[i+1]).$$

2. $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0 \models$ *Run iff* $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \models$ *Schema and* $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, c_0)$ *is well-defined and is a run.*

All the formulae involved in the construction use generalised conjunctions/disjunctions of length linear in the sum of the following values: number of states α, number of transitions β, number of simple loops γ, number of counters n. So, it is easy to check that we only need logarithmic space to build the formulae.

Consequently, we can construct a Presburger formula in logarithmic space that characterises the vectors that can be used to represent runs in a flat counter system. This allows us to quantify over all the runs of a flat counter system starting from a distinguished configuration. The ability to do this lends a useful hand in encoding the model-checking problem in PA without exponential blowup.

5.3 Encoding CTL* formulae using (PA)

We have seen that path schemas and runs can be encoded by tuples of natural numbers whose constraints can be expressed in Presburger arithmetic. Our next aim is to encode a given CTL* formula using a formula in PA. The forthcoming encoding internalises CTL* semantics and a similar idea has been already used in [12] but with much less concise formulae. For each CTL* formula ϕ, we build a PA formula $\mathit{Check}_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$ where the variables $\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}$ and X_0 encode a run as in the formula *Run* and, y encodes a position such that the formula checks whether the CTL* formula is satisfied at the current position. The formula Check_ϕ is defined recursively (Boolean clauses are omitted) as follows:

$$\begin{array}{ll}
\mathit{Check}_{\mathtt{p}} & \stackrel{\text{def}}{=} \exists\, \mathsf{X}\ (\mathit{Conf}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}, \mathsf{X}) \wedge \bigvee_{\{j \mid \mathtt{p} \in \ell(j)\}} \mathsf{x}^1 = j) \\
\mathit{Check}_{\psi(\mathsf{x}_1,\ldots,\mathsf{x}_n)} & \stackrel{\text{def}}{=} \exists\, \mathsf{X}\ (\mathit{Conf}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}, \mathsf{X}) \wedge \psi(\mathsf{x}^2, \ldots, \mathsf{x}^{n+1})) \\
\mathit{Check}_{\mathtt{X}\phi} & \stackrel{\text{def}}{=} \exists\, \mathsf{y}'\ (\mathsf{y}' = \mathsf{y} + 1 \wedge \mathit{Check}_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}')) \\
\mathit{Check}_{\phi\mathtt{U}\phi'} & \stackrel{\text{def}}{=} \exists\, \mathsf{y}''\ (\mathsf{y} \leq \mathsf{y}'' \wedge \mathit{Check}_{\phi'}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}'') \wedge \\
 & \quad \forall\, \mathsf{y}'\ (\mathsf{y} \leq \mathsf{y}' < \mathsf{y}'' \Rightarrow \mathit{Check}_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}'))) \\
\mathit{Check}_{\mathtt{X}^{-1}\phi} & \stackrel{\text{def}}{=} \mathsf{y} > 0 \wedge \exists\, \mathsf{y}'\ (\mathsf{y} = \mathsf{y}' + 1 \wedge \mathit{Check}_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}')) \\
\mathit{Check}_{\phi\mathtt{S}\phi'} & \stackrel{\text{def}}{=} \exists\, \mathsf{y}''\ (\mathsf{y}'' \leq \mathsf{y} \wedge \mathit{Check}_{\phi'}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}'') \wedge \\
 & \quad \forall\, \mathsf{y}'\ (\mathsf{y}'' < \mathsf{y}' \leq \mathsf{y} \Rightarrow \mathit{Check}_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}'))) \\
\mathit{Check}_{\mathtt{E}\phi} & \stackrel{\text{def}}{=} \exists\, \mathsf{Z}'_t\ \exists\, \mathsf{Z}'_p\ \exists\, \mathsf{Z}'_{it}\ (\mathit{Run}(\mathsf{Z}'_t, \mathsf{Z}'_p, \mathsf{Z}'_{it}, \mathsf{X}_0) \wedge \forall \mathsf{y}'(0 \leq \mathsf{y}' \leq \mathsf{y} \Rightarrow \\
 & \quad \forall \mathsf{X}'(\mathit{Conf}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}', \mathsf{X}') \Rightarrow \\
 & \quad \mathit{Conf}(\mathsf{Z}'_t, \mathsf{Z}'_p, \mathsf{Z}'_{it}, \mathsf{X}_0, \mathsf{y}', \mathsf{X}'))) \wedge \\
 & \quad \mathit{Check}_\phi(\mathsf{Z}'_t, \mathsf{Z}'_p, \mathsf{Z}'_{it}, \mathsf{X}_0, \mathsf{y})).
\end{array}$$

Now, we can state the main property concerning the formulae Check_ϕ based on Lemmas 5, 6 and 8.

Lemma 9. *Let* $\mathtt{c}_0 \in \mathbb{N}^{n+1}$, $i \in \mathbb{N}$ *and* $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \in \mathbb{N}^K$ *be such that we have* $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0 \models \mathit{Run}$. *Then, the conditions below are equivalent:*

- $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), i \models \phi$,
- $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i \models \mathit{Check}_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$.

Proof. The proof is by structural induction. Let $\mathtt{c}_0 \in \mathbb{N}^{n+1}$ and $i \in \mathbb{N}$ and $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \in \mathbb{N}^K$ be such that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0 \models \mathit{Run}$. Thus, from Lemma 8, we can conclude that $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)$ is a run in S. We will show that for any i, $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, c_0), i \models \phi$ iff $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i \models \mathit{Check}_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$.

- **Base case 1:** $\phi = \mathtt{p}$. By definition, we have $\mathit{Check}_\mathtt{p}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}) = \exists \mathsf{X}\ \mathit{Conf}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}, \mathsf{X}) \wedge \vee_{\{j \mid \mathtt{p} \in \ell(j)\}} \mathsf{x}^1 = j$. Assume that for some i, $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), i \models \mathtt{p}$. Then $\mathtt{p} \in \ell(\mathtt{c}_i[1])$ with $\mathtt{c}_i = \rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)[i]$. From Lemma 8, we know that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i, \mathtt{c}_i \models \mathit{Conf}$ and $\mathtt{c}_i[1] \in \{j \mid \mathtt{p} \in \ell(j)\}$. Thus, $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i \models \mathit{Check}_\mathtt{p}$.
  Now, assume $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i \models \mathit{Check}_\mathtt{p}$ for some i. Thus, there exists a configuration $\mathtt{c}_i$ such that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i, \mathtt{c}_i \models \mathit{Conf}$ and $\mathtt{c}_i[1] \in \{j \mid \mathtt{p} \in \ell(j)\}$. In that case by Lemma 8, $\mathtt{c}_i = \rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)[i]$ and $\mathtt{p} \in \ell(\mathtt{c}_i[1])$. Hence, $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), i \models \mathtt{p}$.
- **Base case 2:** $\phi = \psi(\mathsf{x}_1, \ldots, \mathsf{x}_n)$. By definition, we have $\mathit{Check}_{\psi(\mathsf{x}_1,\ldots,\mathsf{x}_n)}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}) = \exists \mathsf{X}\ \mathit{Conf}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}, \mathsf{X}) \wedge \psi(\mathsf{x}^2, \ldots, \mathsf{x}^{n+1})$. Assume that for some i, $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), i \models \psi(\mathsf{x}_1, \ldots, \mathsf{x}_n)$. Then $\mathtt{c}_i[2], \ldots \mathtt{c}_i[n+1] \models \psi(\mathsf{x}_1, \ldots, \mathsf{x}_n)$ with $\mathtt{c}_i = \rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)[i]$. Recall that $\mathtt{c}_i[1]$ is a value encoding a control state. From Lemma 8 we know that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i, \mathtt{c}_i \models \mathit{Conf}$ and $\psi(\mathtt{c}_i[2], \ldots, \mathtt{c}_i[n+1])$ is true. Thus, $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i \models \mathit{Check}_{\psi(\mathsf{x}_1,\ldots,\mathsf{x}_n)}$.
  Now, assume $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i \models \mathit{Check}_{\psi(\mathsf{x}_1,\ldots,\mathsf{x}_n)}$ for some i. Thus, there exists a configuration $\mathtt{c}_i$ such that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i, \mathtt{c}_i \models \mathit{Conf}$ and $\psi(\mathtt{c}_i[2], \ldots, \mathtt{c}_i[n+1])$. In that case, by Lemma 8, $\mathtt{c}_i = \rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)[i]$ and $\mathtt{c}_i[2], \ldots, \mathtt{c}_i[n+1] \models \psi(\mathsf{x}_1, \ldots, \mathsf{x}_n)$. Hence, $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), i \models \psi(\mathsf{x}_1, \ldots, \mathsf{x}_n)$.
- **Induction step:** We suppose that for all formulae ϕ such that $|\phi| \leq k$, we have, for all $i \in \mathbb{N}$, $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), i \models \phi$ iff $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i \models \mathit{Check}_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$. We will prove the same for the case when $|\phi| = k+1$.
  - $\mathtt{X}\phi$*:* First we assume that $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), i \models \mathtt{X}\phi$ for some i. By definition of $\models$, we have $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), i+1 \models \phi$. By the induction hypothesis, $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i+1 \models \mathit{Check}_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$. We can hence deduce that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i \models \mathit{Check}_{\mathtt{X}\phi}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$.
    Now, assume $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i \models \mathit{Check}_{\mathtt{X}\phi}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$ for some i. By definition of $\mathit{Check}_{\mathtt{X}\phi}$, we have that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i+1 \models \mathit{Check}_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$. By the induction hypothesis, $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), i+1 \models \phi$. By definition of $\models$, we get $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), i \models \mathtt{X}\phi$.
  - $\mathtt{X}^{-1}\phi$*:* First we assume that $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), i \models \mathtt{X}^{-1}\phi$ for some i. By definition of $\models$, we have $i > 0$ $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), i-1 \models \phi$. By the induction hypothesis, $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i-1 \models \mathit{Check}_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$. We can hence deduce that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i \models \mathit{Check}_{\mathtt{X}^{-1}\phi}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$.
    Now, assume $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i \models \mathit{Check}_{\mathtt{X}^{-1}\phi}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$ for some i. By definition of $\mathit{Check}_{\mathtt{X}^{-1}\phi}$, we have that $i > 0$ and $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i-1 \models$

$Check_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$. By the induction hypothesis, $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)$, $i-1 \models \phi$. By definition of $\models$, we get $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), i \models \mathtt{X}^{-1}\phi$.

- $\phi \mathtt{U} \phi'$: First we assume that $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), i \models \phi \mathtt{U} \phi'$ for some i. By definition of $\models$, there exists $j \geq i$ such that $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), j \models \phi'$ and for all $i \leq l < j$, $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), l \models \phi$. By the induction hypothesis, we get $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, j \models Check_{\phi'}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$ and for all $i \leq l < j$, $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, l \models Check_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$. We can hence deduce that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i \models Check_{\phi \mathtt{U} \phi'}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$.
  Now, assume $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i \models Check_{\phi \mathtt{U} \phi'}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$ for some i. By definition of $Check_{\phi \mathtt{U} \phi'}$, we have that there exists $i \leq j$ such that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, j \models Check_{\phi'}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$ and for all $i \leq l < j$, $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, l \models Check_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$. By the induction hypothesis, we obtain $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), j \models \phi'$ and for all $i \leq l < j$, $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), l \models \phi$. By definition of $\models$, we get that, $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), i \models \phi \mathtt{U} \phi'$.
- $\phi \mathtt{S} \phi'$: First we assume that $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), i \models \phi \mathtt{S} \phi'$ for some i. By satisfaction relation, there exists $j \leq i$ such that $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), j \models \phi'$ and for all $j < l \leq i$, $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), l \models \phi$. By induction hypothesis, we get $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, j \models Check_{\phi'}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$ and for all $j < l \leq i$, $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, l \models Check_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$. Hence, we conclude that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i \models Check_{\phi \mathtt{S} \phi'}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$.
  Assume now $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i \models Check_{\phi \mathtt{S} \phi'}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$ for some i. By definition of $Check_{\phi \mathtt{S} \phi'}$, we have that there exists $j \leq i$ such that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, j \models Check_{\phi'}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$ and for all $j < l \leq i$, $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, l \models Check_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$. By the induction hypothesis, we obtain $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), j \models \phi'$ and for all $j < l \leq i$, $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), l \models \phi$. By definition of $\models$, we get that, $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), i \models \phi \mathtt{S} \phi'$.
- $\mathtt{E}\phi$: Let us assume that $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), i \models \mathtt{E}\phi$ for some i. By definition of the satisfaction relation, there exists a run ρ' such that $\rho'_{\leq i} = \rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)_{\leq i}$ and $\rho', i \models \phi$. By Lemma 5, we know that there exists an iterated path schema $\langle P, \mathtt{m} \rangle$ such that $trans(\rho') = trans(P, \mathtt{m})$. From Lemma 6, we get that there exist vectors $\mathtt{v}'_t, \mathtt{v}'_p, \mathtt{v}'_{it}$ such that $trans(\rho') = trans(P_{\mathtt{v}'_t,\mathtt{v}'_p}, \mathtt{m}_{\mathtt{v}'_{it}})$ and ρ' can be represented as $\rho'(P_{\mathtt{v}'_t,\mathtt{v}'_p}, \mathtt{m}_{\mathtt{v}'_{it}}, \mathtt{c}_0)$. Thus, $\rho'(P_{\mathtt{v}'_t,\mathtt{v}'_p}, \mathtt{m}_{\mathtt{v}'_{it}}, \mathtt{c}_0), i \models \phi$. By the induction hypothesis, we have

$$\mathtt{v}'_t, \mathtt{v}'_p, \mathtt{v}'_{it}, \mathtt{c}_0, i \models Check_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}).$$

  Moreover, since $\rho'(P_{\mathtt{v}'_t,\mathtt{v}'_p}, \mathtt{m}'_{\mathtt{v}'_{it}}, \mathtt{c}_0)$ is a run, by Lemma 8 we get $\mathtt{v}'_t, \mathtt{v}'_p, \mathtt{v}'_{it}, \mathtt{c}_0 \models Run$. Finally since we have that $\rho'_{\leq i} = \rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)_{\leq i}$, we deduce that for $j \in [0, i]$, we have $\rho'[j] = \rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)[j]$. Hence, using the result of Lemma 8 concerning the formula $Conf$, we conclude that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i \models Check_{\mathtt{E}\phi}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$.
  Now, assume that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i \models Check_{\mathtt{E}\phi}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$. Thus, by definition of $Check_{\mathtt{E}\phi}$ and using Lemma 8, we obtain that there exist $\mathtt{v}'_t, \mathtt{v}'_p, \mathtt{v}'_{it}$ such that $\rho(P_{\mathtt{v}'_t,\mathtt{v}'_p}, \mathtt{m}_{\mathtt{v}'_{it}}, \mathtt{c}_0)$ is a run and $\mathtt{v}'_t, \mathtt{v}'_p, \mathtt{v}'_{it}, \mathtt{c}_0, i \models Check_\phi$. Furthermore, the formula $Check_{\mathtt{E}\phi}$ allows us as well to state that for all $j \in [0, i]$, we have $\rho(P_{\mathtt{v}'_t,\mathtt{v}'_p}, \mathtt{m}_{\mathtt{v}'_{it}}, \mathtt{c}_0)[i] = \rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)[i]$.

By the induction hypothesis, $\rho(P_{\mathtt{v}'_t,\mathtt{v}'_p}, \mathtt{m}'_{\mathtt{v}'_{it}}, \mathtt{c}_0), i \models \phi$. So there is a run ρ' such that $\rho'_{\leq i} = \rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)_{\leq i}$ and $\rho', i \models \phi$. By definition of the satisfaction relation, we get $\rho'(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), i \models \mathtt{E}\phi$.

- The other cases with Boolean connectives can be proved in a similar manner.

□

In view of the previous results, we are in a position to construct a Presburger formula pertaining to our original aim of encoding the model-checking problem MC(CTL*, `FlatCS`). Given a flat counter system S, a configuration $\mathtt{c}_0$ and a formula ϕ in CTL*, we construct the following Presburger arithmetic formula ψ as

$$\psi = \exists \mathsf{Z}_t\ \exists \mathsf{Z}_p\ \exists \mathsf{Z}_{it}\ \exists \mathsf{X}_0\ (\mathsf{X}_0 = \mathtt{c}_0 \wedge \mathit{Run}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0) \\ \wedge \mathit{Check}_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, 0)).$$

Then, Lemmas 8 and 9 allow us to get the following property.

Lemma 10. *The formula ψ is satisfiable iff $S, \mathtt{c}_0 \models \phi$.*

Proof. Suppose that ψ is satisfiable. Then there exist $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}$ such that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, 0 \models \mathit{Check}_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y})$ and $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0 \models \mathit{Run}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0)$. By Lemma 8, since $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0 \models \mathit{Run}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0)$, we know that $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)$ is a run starting at $\mathtt{c}_0$ and using Lemma 9, we have that $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0), 0 \models \phi$. Hence $S, \mathtt{c}_0 \models \phi$.

Suppose now that $S, \mathtt{c}_0 \models \phi$. Hence there is a run ρ starting at $\mathtt{c}_0$ such that $\rho, 0 \models \phi$. Thanks to Lemmas 5 and 6, we know that there exist $\mathtt{v}_t$, $\mathtt{v}_p$ and $\mathtt{v}_{it}$ in $\mathbb{N}^K$ such that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \models \mathit{Schema}$ and $\rho = \rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)$. From Lemma 8 and Lemma 9, we get $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0 \models \mathit{Run}$ and $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, 0 \models \mathit{Check}_\phi$. So, ψ is satisfiable. □

This allows us to conclude the main result of this section.

Theorem 11. *There is a logarithmic-space reduction from MC(*CTL*, `FlatCS`*) to SAT(PA).*

Equipped with Theorem 11 and Theorem 3 and the fact that $\mathrm{CTL}_{\mathtt{EF}}$ is a subclass of CTL*, we can conclude that the model-checking problem of flat counter systems with CTL* is equivalent to SAT(PA), modulo logarithmic-space reductions.

6 Global Model-Checking

The method we have used so far is robust enough to draw more conclusions. As shown below, it is also possible to construct a formula in PA that captures the set of (initial) counter values from which there is an infinite run satisfying a given CTL* formula. The global model-checking problem over flat counter systems with CTL* (written GMC(CTL*, `FlatCS`)) is the following:

Input: A flat counter system S with n counters, a control state q, a CTL* formula ϕ.
Output: A Presburger formula $\Psi(\mathsf{z}_1, \ldots, \mathsf{z}_n)$ such that

$$[\![\Psi]\!] = \{\mathtt{v} \in \mathbb{N}^n \ \mid \ S, \langle q, \mathtt{v}\rangle \models \phi\}.$$

It is actually quite easy to construct the Presburger formula for the global model-checking problem with CTL* as the model-checking problem itself is encoded by a Presburger formula. In the previous section, we saw that given a flat counter system S, a configuration c_0 and a formula ϕ in CTL*, we construct the following Presburger arithmetic formula ψ as described below.

$$\psi = \exists \mathsf{Z}_t \ \exists \mathsf{Z}_p \ \exists \mathsf{Z}_{it} \ \exists \mathsf{X}_0 \ (\mathsf{X}_0 = \mathsf{c}_0 \wedge \mathit{Run}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0) \\ \wedge \mathit{Check}_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, 0)).$$

For the global model-checking problem if we suppose that the control states of the counter systems are encoded as natural numbers, given a flat counter system S, a control state q and a formula ϕ in CTL*, we construct the Presburger formula $\Psi(\mathsf{z}_1, \ldots, \mathsf{z}_n)$ as

$$\Psi(\mathsf{z}_1, \ldots, \mathsf{z}_n) = \exists \mathsf{Z}_t \ \exists \mathsf{Z}_p \ \exists \mathsf{Z}_{it} \ \exists \mathsf{X}_0 \ (\mathsf{x}_0^1 = q \wedge \bigwedge_{i=1}^{n} \mathsf{x}_0^i = \mathsf{z}_i \wedge \mathit{Run}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0) \\ \wedge \mathit{Check}_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, 0)).$$

As a consequence of Lemma 10 and other previous results, we get the following characterisation.

Lemma 12. *For all* $\mathtt{v} \in \mathbb{N}^n$*, we have* $\mathtt{v} \models \Psi(\mathsf{z}_1, \ldots, \mathsf{z}_n)$ *iff there exists a run* ρ *in* S *starting from* $\langle q, \mathtt{v}\rangle$ *such that* $\rho, 0 \models \phi$*.*

As a conclusion, our method provides also a means to synthetize Presburger formulae for solving the global model-checking problem.

7 MC(CTL*, `FlatCS`) subproblems and PA fragments

The construction and translation presented in the previous sections entail that the model-checking problem for CTL* over flat counter systems and the satisfiability problem for PA are inter-reducible . Though this provides a nice complexity characterization of the model-checking problem for full CTL* logic over flat counter systems, our translation allows us to provide some tighter bounds for CTL* fragments. In [20], the author characterizes the precise complexity of PA fragments and shows equivalences with classes from the weak-Exp hierarchy. In this section, we establish a comparison between PA fragments as those considered in [20] and the model-checking problem for fragments of CTL* over flat counter systems.

For subclasses of PA, we borrow the notations used in [20] and denote by PA(i), the Σ_i-fragment of Presburger arithmetic restricted to i quantifier alternations beginning with an existential quantifier. Let Σ_i^{Exp} be the ith level of the

weak-Exp hierarchy which is analogous to the polynomial-time hierarchy but built with the help of ExpTime (see e.g. [24]). The following result, established in [20], is stated below.

Theorem 13. *For any fixed $i > 0$, the PA$(i+1)$ fragment of Presburger arithmetic is Σ_i^{Exp}-complete.*

In order to analyze the complexity of our model-checking problem and to use this latter result on PA, we need to focus on a restriction of CTL* where the atomic propositions $\psi(\mathtt{x}_1, \ldots, \mathtt{x}_n)$ that describe the counter values belong to the linear fragment of PA (i.e., ψ is quantifier-free). We denote by CTL*$_{\mathrm{LIA}}$ this fragment of CTL*.

Now, we aim at analyzing the PA formulae generated by the translation of a specific instance of the model-checking problem as presented in Section 5. Since we are interested in the quantifier alternation of the produced formulae, negation may increase the number of quantifier alternations. In order to simplify the analysis, without any of loss of generality, we can assume that the negations are pushed until the atomic propositions and we consequently need to extend the syntax of CTL* with the disjunction operator, the universal path quantifiers, $\mathtt{A}\phi \stackrel{\text{def}}{=} \neg\mathtt{E}\neg\phi$, the release operator $\phi\mathtt{R}\phi' \stackrel{\text{def}}{=} \neg(\neg\phi\mathtt{U}\neg\psi)$ and the past release operator $\phi\mathtt{R}^{-1}\phi' \stackrel{\text{def}}{=} \neg(\neg\phi\mathtt{S}\neg\psi)$. Thanks to these dual operators, we can restrict ourselves to CTL*$_{\mathrm{LIA}}$ formulae in negation normal form (negation can only occur in front of atomic propositions).

We consider now a flat counter system S, a configuration $\mathtt{c}$ and a formula ϕ in negation normal form belonging to CTL*$_{\mathrm{LIA}}$. We will compute an integer K such that the translation of the instance of MC(CTL*, `FlatCS`), provided in the previous section, yields a PA formula in the fragment PA(K). In fact, we will see that K depends on the joint quantifier and temporal depth of the CTL*$_{\mathrm{LIA}}$ formula ϕ, denoted by $qtd(\phi)$ and defined inductively as follows :

- $qtd(\mathtt{p}) = qtd(\psi(\mathtt{x}_1, \ldots, \mathtt{x}_n)) = 0$,
- $qtd(\neg\phi) = qtd(\phi)$, $qtd(\mathtt{X}\phi) = 1 + qtd(\phi)$,
- $qtd(\phi\mathtt{U}\phi') = 1 + \max(qtd(\phi), qtd(\phi'))$, $qtd(\phi\mathtt{R}\phi') = 1 + \max(qtd(\phi), qtd(\phi'))$,
- $qtd(\mathtt{X}^{-1}\phi) = 1 + qtd(\phi)$, $qtd(\phi\mathtt{S}\phi') = 1 + \max(qtd(\phi), qtd(\phi'))$,
- $qtd(\phi\mathtt{R}^{-1}\phi') = 1 + \max(qtd(\phi), qtd(\phi'))$,
- $qtd(\mathtt{E}\phi) = 1 + qtd(\phi)$ and $qtd(\mathtt{A}\phi) = 1 + qtd(\phi)$.

Now, let us analyse the formula produced in Section 5. We recall that it has the following shape:

$$\psi = \exists \mathsf{Z}_t\ \exists \mathsf{Z}_p\ \exists \mathsf{Z}_{it}\ \exists \mathsf{X}_0\ (\mathsf{X}_0 = \mathtt{c}_0 \wedge \mathit{Run}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0) \\ \wedge \mathit{Check}_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, 0)).$$

The fragment in which this formula belongs to depends on the shape of the formulas $\mathit{Run}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0)$ and $\mathit{Check}_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, 0)$. We begin our analysis with the easier formula $\mathit{Run}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0)$. By definition,

$$\mathit{Run}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0) \stackrel{\text{def}}{=} \forall\ \mathsf{y}\ \exists\ \mathsf{X}\ \mathit{Conf}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}, \mathsf{X}).$$

By construction *Conf* belongs to the existential fragment of PA and hence *Run* belongs to PA(3) (by definition PA(i) starts with an existential quantifier).

Let us now recall how the $Check_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, 0)$ is inductively defined :

$$
\begin{array}{ll}
Check_{\mathtt{p}} & \stackrel{\text{def}}{=} \exists\, \mathsf{X}\ (Conf(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}, \mathsf{X}) \wedge \bigvee_{\{j \mid \mathtt{p} \in \ell(j)\}} \mathsf{x}^1 = j) \\
Check_{\psi(\mathsf{x}_1,\ldots,\mathsf{x}_n)} & \stackrel{\text{def}}{=} \exists\, \mathsf{X}\ (Conf(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}, \mathsf{X}) \wedge \psi(\mathsf{x}^2, \ldots, \mathsf{x}^{n+1})) \\
Check_{\mathtt{X}\phi} & \stackrel{\text{def}}{=} \exists\, \mathsf{y}'\ (\mathsf{y}' = \mathsf{y} + 1 \wedge Check_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}')) \\
Check_{\phi\mathtt{U}\phi'} & \stackrel{\text{def}}{=} \exists\, \mathsf{y}''\ (\mathsf{y} \leq \mathsf{y}'' \wedge Check_{\phi'}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}'') \wedge \\
 & \quad \forall\, \mathsf{y}'\ (\mathsf{y} \leq \mathsf{y}' < \mathsf{y}'' \Rightarrow Check_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}'))) \\
Check_{\mathtt{X}^{-1}\phi} & \stackrel{\text{def}}{=} \mathsf{y} > 0 \wedge \exists\, \mathsf{y}'\ (\mathsf{y} = \mathsf{y}' + 1 \wedge Check_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}')) \\
Check_{\phi\mathtt{S}\phi'} & \stackrel{\text{def}}{=} \exists\, \mathsf{y}''\ (\mathsf{y}'' \leq \mathsf{y} \wedge Check_{\phi'}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}'') \wedge \\
 & \quad \forall\, \mathsf{y}'\ (\mathsf{y}'' < \mathsf{y}' \leq \mathsf{y} \Rightarrow Check_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}'))) \\
Check_{\mathtt{E}\phi} & \stackrel{\text{def}}{=} \exists\, \mathsf{Z}'_t\ \exists\, \mathsf{Z}'_p\ \exists\, \mathsf{Z}'_{it}\ (Run(\mathsf{Z}'_t, \mathsf{Z}'_p, \mathsf{Z}'_{it}, \mathsf{X}_0) \wedge \forall \mathsf{y}'(0 \leq \mathsf{y}' \leq \mathsf{y} \Rightarrow \\
 & \quad \forall \mathsf{X}'(Conf(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}', \mathsf{X}') \Rightarrow \\
 & \quad Conf(\mathsf{Z}'_t, \mathsf{Z}'_p, \mathsf{Z}'_{it}, \mathsf{X}_0, \mathsf{y}', \mathsf{X}'))) \wedge \\
 & \quad Check_\phi(\mathsf{Z}'_t, \mathsf{Z}'_p, \mathsf{Z}'_{it}, \mathsf{X}_0, \mathsf{y})).
\end{array}
$$

In order to complete our analysis we need as well to provide the translation for the negation of atomic propositions and for the newly introduced operators $\mathtt{A}$, $\mathtt{R}$ and $\mathtt{R}^{-1}$:

$$
\begin{array}{ll}
Check_{\neg\mathtt{p}} & \stackrel{\text{def}}{=} \exists\, \mathsf{X}\ (Conf(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}, \mathsf{X}) \wedge \bigvee_{\{j \mid \mathtt{p} \notin \ell(j)\}} \mathsf{x}^1 = j) \\
Check_{\neg\psi(\mathsf{x}_1,\ldots,\mathsf{x}_n)} & \stackrel{\text{def}}{=} \exists\, \mathsf{X}\ (Conf(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}, \mathsf{X}) \wedge \neg\psi(\mathsf{x}^2, \ldots, \mathsf{x}^{n+1})) \\
Check_{\phi\mathtt{R}\phi'} & \stackrel{\text{def}}{=} (\exists\, \mathsf{y}''\ (\mathsf{y} \leq \mathsf{y}'' \wedge Check_\phi(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}'') \wedge \\
 & \quad \forall\, \mathsf{y}'\ (\mathsf{y} \leq \mathsf{y}' \leq \mathsf{y}'' \Rightarrow Check_{\phi'}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}')))) \vee \\
 & \quad \forall\, \mathsf{y}'\ (\mathsf{y} \leq \mathsf{y}' \Rightarrow Check_{\phi'}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}')) \\
Check_{\phi\mathtt{R}^{-1}\phi'} & \stackrel{\text{def}}{=} (\exists\, \mathsf{y}''\ (\mathsf{y}'' \leq \mathsf{y} \wedge Check_{\phi'}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}'') \wedge \\
 & \quad \forall\, \mathsf{y}'\ (\mathsf{y}'' \leq \mathsf{y} \leq \mathsf{y} \Rightarrow Check_{\phi'}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}')))) \vee \\
 & \quad \forall\, \mathsf{y}'\ (\mathsf{y}' \leq \mathsf{y} \Rightarrow Check_{\phi'}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}')) \\
Check_{\mathtt{A}\phi} & \stackrel{\text{def}}{=} \forall\, \mathsf{Z}'_t\ \forall\, \mathsf{Z}'_p\ \forall\, \mathsf{Z}'_{it}\ (Run(\mathsf{Z}'_t, \mathsf{Z}'_p, \mathsf{Z}'_{it}, \mathsf{X}_0) \wedge \forall \mathsf{y}'(0 \leq \mathsf{y}' \leq \mathsf{y} \Rightarrow \\
 & \quad \forall \mathsf{X}'(Conf(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}', \mathsf{X}') \Rightarrow \\
 & \quad Conf(\mathsf{Z}'_t, \mathsf{Z}'_p, \mathsf{Z}'_{it}, \mathsf{X}_0, \mathsf{y}', \mathsf{X}'))) \Rightarrow \\
 & \quad Check_\phi(\mathsf{Z}'_t, \mathsf{Z}'_p, \mathsf{Z}'_{it}, \mathsf{X}_0, \mathsf{y})).
\end{array}
$$

The proof for the correctness of the extended translation is similar to the one for Lemma 9. Now, we can perform a case analysis to determine the number of quantifier alternations in the formula $Check_\phi$ expressed in terms of $qtd(\phi)$ and we obtain the following upper bound.

Lemma 14. *$Check_\phi$ belongs to* PA($2.qtd(\phi) + 3$).

Proof. The proof is by structural induction. For the base case, it is clear that $Check_{\mathtt{p}}$, $Check_{\neg\mathtt{p}}$, $Check_{\psi(\mathsf{x}_1,\ldots,\mathsf{x}_n)}$ and $Check_{\neg\psi(\mathsf{x}_1,\ldots,\mathsf{x}_n)}$ are in PA(3).

For the induction step, we assume that $Check_\phi$ and $Check_{\phi'}$ belong respectively to $\mathrm{PA}(2qtd(\phi)+3)$ and $\mathrm{PA}(2qtd(\phi')+3)$. Then $Check_{\mathtt{X}\phi}$ and $Check_{\mathtt{X}^{-1}\phi}$ belongs as well to $\mathrm{PA}(2qtd(\phi)+3)$. Furthermore, we have $Check_{\phi\mathtt{U}\phi'}$ belongs to $\mathrm{PA}(max(2qtd(\phi)+3,(2qtd(\phi')+3)+2)$ and since $qtd(\phi\mathtt{U}\phi') = 1+\max(qtd(\phi), qtd(\phi'))$, we have that $Check_{\phi\mathtt{U}\phi'}$ belongs to $\mathrm{PA}(2qtd(\phi\mathtt{U}\phi')+3)$. The exact same reasoning holds for $\mathtt{S}$, $\mathtt{R}$ and $\mathtt{R}^{-1}$. Finally we have that $Check_{\mathtt{E}\phi}$ belongs to $\mathrm{PA}(max(3,(2qtd(\phi)+3)+2)$ which is in $\mathrm{PA}(2(qtd(\mathtt{E}\phi)+3))$. Similarly, $Check_{\mathtt{A}\phi}$ belongs to $\mathrm{PA}(max(3,2(qtd(\phi)+3)+2)$ which is in $\mathrm{PA}(2qtd(\mathtt{A}\phi)+3)$.

If, for $k \in \mathbb{N}$, we denote by $\mathrm{CTL}^*_{\mathrm{LIA}}(k)$ the set of formulas ϕ of $\mathrm{CTL}^*_{\mathrm{LIA}}$ such that $qtd(\phi) \leq k$, the previous lemma together with the previous observations and Theorem 13 allow us to state the result below.

Theorem 15. *$MC(\mathrm{CTL}^*_{\mathrm{LIA}}(k), \mathtt{FlatCS}))$ is reducible to the satisfiability problem for $PA(2k+3)$ and is in $\Sigma^{\mathrm{Exp}}_{2k+2}$*

Note that by analyzing the formula built in the proof of Theorem 3 which shows that the satisfiability of PA can be reduced to the model-checking of $\mathrm{CTL^*}_{\mathrm{LIA}}$ over flat counter systems, we are able to obtain a lower bound. In fact we observe that the $\mathrm{CTL^*}_{\mathrm{LIA}}$ formula ψ built from a formula of $\mathrm{PA}(k)$ for $k \in \mathbb{N}$ is such that $qtd(\psi) = 2.k+2$. The following result can then be deduced.

Theorem 16. *For $k \geq 4$, $MC(\mathrm{CTL}^*_{\mathrm{LIA}}(k), \mathtt{FlatCS})$is $\Sigma^{\mathrm{Exp}}_{\lfloor k/2 \rfloor -2}$-hard.*

8 Conclusion

We have been able to characterise the computational complexity for MC(CTL^*, $\mathtt{FlatCS}$) by showing that the problem is equivalent to SAT(PA) (modulo logarithmic-space reductions). The lower bound is obtained by considering a quite strong restriction (no arithmetical constraints in formulae, the only temporal operator is $\mathtt{EF}$, guards on transitions are simple linear constraints). By contrast, the restriction of the problem to LTL formulae is known to be NP-complete [9,13] when guards are in the linear fragment and the restriction of the problem to formulae in $\mathrm{CTL}_{\mathtt{EF}}$ is also equivalent to SAT(PA). We have proposed a new way for encoding runs in flat counter systems using Presburger arithmetic formulae, but without any exponential blow up, which allows us to get a precise complexity characterisation. It remains open to determine which extensions of CTL^* on flat counter systems preserve decidability, if not an efficient translation into PA (typically for logics between CTL^* and the modal mu-calculus).

Acknowledgments: We would like to thank the anonymous referees for their remarks and suggestions that help us to improve the quality of the paper.

References

1. C. Barrett, C. Conway, M. Deters, L. Hadarean, D. Jovanovic, T. King, A. Reynolds, and C. Tinelli. CVC4. In *CAV'11*, volume 8606 of *Lecture Notes in Computer Science*, pages 171–177. Springer, 2011.

2. L. Berman. The complexity of logical theories. *Theoretical Computer Science*, 11:71–78, 1980.
3. M. Bersani and S. Demri. The complexity of reversal-bounded model-checking. In *FROCOS'11*, volume 6989 of *Lecture Notes in Artificial Intelligence*, pages 71–86. Springer, 2011.
4. B. Boigelot. *Symbolic methods for exploring infinite state spaces*. PhD thesis, Université de Liège, 1998.
5. M. Bozga, R. Iosif, and F. Konecny. Safety problems are NP-complete for flat integer programs with octagonal loops. In *VMCAI'14*, volume 8318 of *Lecture Notes in Computer Science*, pages 242–261. Springer, 2014.
6. V. Bruyère, E. Dall'Olio, and J. Raskin. Durations, parametric model-checking in timed automata with Presburger arithmetic. In *STACS'03*, volume 2607 of *Lecture Notes in Computer Science*, pages 687–698. Springer, 2003.
7. H. Comon and Y. Jurski. Multiple counter automata, safety analysis and Presburger Arithmetic. In *CAV'98*, volume 1427 of *Lecture Notes in Computer Science*, pages 268–279. Springer, 1998.
8. L. de Moura and N. Björner. Z3: An Efficient SMT Solver. In *TACAS'08*, volume 4963 of *Lecture Notes in Computer Science*, pages 337–340. Springer, 2008.
9. S. Demri, A. Dhar, and A. Sangnier. Taming Past LTL and Flat Counter Systems. In *IJCAR'12*, volume 7364 of *Lecture Notes in Artificial Intelligence*, pages 179–193. Springer, 2012.
10. S. Demri, A. Dhar, and A. Sangnier. On the complexity of verifying regular properties on flat counter systems. In *ICALP'13*, volume 7966 of *Lecture Notes in Computer Science*, pages 162–173. Springer, 2013.
11. S. Demri, A. Dhar, and A. Sangnier. Equivalence between model-checking flat counter systems and Presburger arithmetic. In *RP'14*, volume 8762 of *Lecture Notes in Computer Science*, pages 85–97. Springer, 2014.
12. S. Demri, A. Finkel, V. Goranko, and G. van Drimmelen. Model-checking CTL* over flat Presburger counter systems. *Journal of Applied Non-Classical Logics*, 20(4):313–344, 2010.
13. A. Dhar. *Algorithms for model-checking flat counter systems*. PhD thesis, Université Paris Diderot, December 2014.
14. E. Emerson and J. Halpern. 'sometimes' and 'not never' revisited: on branching versus linear time temporal logic. *Journal of the ACM*, 33:151–178, 1986.
15. E. Emerson and C.-L. Lei. Modalities for model checking: Branching time logic strikes back. *Sci. Comput. Program.*, 8(3):275–306, 1987.
16. A. Finkel and J. Leroux. How to compose Presburger accelerations: Applications to broadcast protocols. In *FST&TCS'02*, volume 2256 of *Lecture Notes in Computer Science*, pages 145–156. Springer, 2002.
17. L. Fribourg and H. Olsén. Proving safety properties of infinite state systems by compilation into Presburger arithmetic. In *CONCUR'97*, volume 1243 of *Lecture Notes in Computer Science*, pages 213–227. Springer, 1997.
18. S. Göller, C. Haase, J. Ouaknine, and J. Worrell. Branching-time model checking of parametric one-counter automata. In *FoSSaCS'12*, volume 7213 of *Lecture Notes in Computer Science*, pages 406–420. Springer, 2012.
19. S. Göller and M. Lohrey. Branching-time model checking of one-counter processes and timed automata. *SIAM J. Comput.*, 42(3):884–923, 2013.
20. C. Haase. Subclasses of Presburger arithmetic and the weak EXP hierarchy. In *CSL-LICS'14*, pages 47:1–47:10. ACM, 2014.

21. C. Haase, S. Kreutzer, J. Ouaknine, and J. Worrell. Reachability in succinct and parametric one-counter automata. In *CONCUR'09*, volume 5710 of *Lecture Notes in Computer Science*, pages 369–383. Springer, 2009.
22. P. Habermehl. On the complexity of the linear-time mu-calculus for Petri nets. In *ICATPN'97*, volume 1248 of *Lecture Notes in Computer Science*, pages 102–116. Springer, 1997.
23. M. Hague and A. W. Lin. Model checking recursive programs numeric data types. In *CAV'11*, volume 6806 of *Lecture Notes in Computer Science*, pages 743–759. Springer, 2011.
24. L. Hemachandra. The strong exponential hierarchy collapses. *JCSS*, 39(3):299–322, 1989.
25. O. Ibarra. Reversal-bounded multicounter machines and their decision problems. *Journal of the ACM*, 25:116–133, 1978.
26. S. R. Kosaraju. Decidability of reachability in vector addition systems (preliminary version). In *STOC'82*, pages 267–281, 1982.
27. F. Laroussinie, N. Markey, and P. Schnoebelen. Model checking CTL^+ and FCTL is hard. In *FOSSACS'01*, volume 2030 of *Lecture Notes in Computer Science*, pages 318–331. Springer, 2001.
28. A. Lechner, R. Mayr, J. Ouaknine, A. Pouly, and J. Worrell. Model Checking Flat Freeze LTL on One-Counter Automata. In *CONCUR'16*, volume 59, pages 29:1–29:14. LIPIcs, 2016.
29. J. Leroux. Presburger counter machines. *Habilitation thesis, U. of Bordeaux*, 2012.
30. J. Leroux and G. Point. TaPAS: The Talence Presburger Arithmetic Suite. In *TACAS'09*, volume 5505 of *Lecture Notes in Computer Science*, pages 182–185. Springer, 2009.
31. J. Leroux and G. Sutre. On flatness for 2-dimensional vector addition systems with states. In *CONCUR*, volume 3170 of *LNCS*, pages 402–416. Springer, 2004.
32. J. Leroux and G. Sutre. Flat counter automata almost everywhere! In *ATVA'05*, volume 3707 of *Lecture Notes in Computer Science*, pages 489–503. Springer, 2005.
33. E. Mayr. Persistence of vector replacement systems is decidable. *Acta Informatica*, 15:309–318, 1981.
34. M. Minsky. *Computation, Finite and Infinite Machines.* Prentice Hall, 1967.
35. C. Papadimitriou. *Computational complexity.* Addison-Wesley, 1994.
36. M. Presburger. Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In *Comptes Rendus du premier congrès de mathématiciens des Pays Slaves, Warszawa*, pages 92–101, 1929.
37. N. Suzuki and D. Jefferson. Verification Decidability of Presburger Array Programs. *Journal of the ACM*, 27(1):191–205, 1980.

A Proof of Lemma 2

Proof. First, we show that every maximal run in S' is infinite. For every maximal run ρ in S, we write $\texttt{maximize}(\rho)$ to denote the maximal run in S' defined as follows:

– if ρ is infinite, then $\texttt{maximize}(\rho) \stackrel{\text{def}}{=} \rho$,

– otherwise, we have:

$$\texttt{maximize}(\rho) \stackrel{\text{def}}{=} \rho \xrightarrow{\delta_{\alpha+1}} \langle q_\perp, \mathtt{v}_\alpha\rangle \xrightarrow{\delta_\perp} \langle q_\perp, \mathtt{v}_\alpha\rangle \xrightarrow{\delta_\perp} \cdots$$

where
- $|\rho| = \alpha$,
- $\rho[\alpha] = \langle q_\alpha, \mathtt{v}_\alpha\rangle$,
- $\delta_\perp = \langle q_\perp, \top, \mathtt{0}, q_\perp\rangle$ and
- $\delta_{\alpha+1} = \langle q_\alpha, \neg(\vee_{source(\delta)=q_\alpha} guard(\delta)), \mathtt{0}, q_\perp\rangle$.

Since $\langle q_\alpha, \mathtt{v}_\alpha\rangle$ is a deadlock configuration, we have $\mathtt{v}_\alpha \not\models guard(\delta)$ for every δ such that $source(\delta) = q_\alpha$. We conclude that $\mathtt{v}_\alpha \models \neg(\vee_{source(\delta)=q_\alpha} guard(\delta))$ and hence the only possible configuration after $\langle q, \mathtt{v}\rangle$ is $\langle q_\perp, \mathtt{v}\rangle$. So, for all such finite runs this is the only way to get an infinite maximal run extending ρ.

In other words, if we assume that $\rho_\perp = \langle q_\perp, \mathtt{v}_\alpha\rangle \xrightarrow{\delta_\perp} \langle q_\perp, \mathtt{v}_\alpha\rangle \xrightarrow{\delta_\perp} \cdots$. We can define the function $\texttt{maximize}(\cdot)$ for any run ρ of S as following

$$\texttt{maximize}(\rho) \stackrel{\text{def}}{=} \begin{cases} \rho & \text{if } \rho \text{ is infinite} \\ \rho \xrightarrow{\delta_{\alpha+1}} \rho_\perp & |\rho| = \alpha, \text{ for some } \alpha \in \mathbb{N} \end{cases}$$

So, clearly, for any run ρ in S there exists an unique run ρ' in S' such that $\rho' = \texttt{maximize}(\rho)$. Hence, there is a bijection between the set of maximal runs in S and the set of maximal runs in S' that start by a state in Q. Every maximal run in S' is infinite. Below, we also use the inverse map $\texttt{maximize}^{-1}(\cdot)$ that returns a maximal run in S.

By structural induction, we show that for all ρ in S and $i \leq |\rho|$, we have $\rho, i \models \phi$ iff $\texttt{maximize}(\rho), i \models \mathfrak{t}(\phi)$. As a consequence, for all configurations c in S and for all formulae ϕ in CTL* [resp. in $\text{CTL}_{\mathtt{EF}}$], we have $S, c \models \phi$ iff $S', c \models \mathfrak{t}(\phi)$. By construction, S' is non-blocking and, S' and ϕ' can be built in logarithmic space in the size of S and ϕ.

– **Base case:** Let ρ be a run in S and $i \leq |\rho|$. Since we have $i \leq |\rho|$ we deduce that $\rho[i] = \texttt{maximize}(\rho)[i]$. As $\mathfrak{t}(\cdot)$ is the identity for atomic formulae $\mathtt{p}$ and $\psi(\mathtt{x}_1, \cdots, \mathtt{x}_n)$, we conclude that $\rho, i \models \mathtt{p}$ [resp. $\rho, i \models \psi(\mathtt{x}_1, \cdots, \mathtt{x}_n)$] iff $\rho', i \models \mathfrak{t}(\mathtt{p})$ [resp. $\rho', i \models \mathfrak{t}(\psi(\mathtt{x}_1, \cdots, \mathtt{x}_n))$].
– **Induction step:** We assume that for all formulae ϕ of size at most $k-1$, for all runs ρ and for all $i \leq |\rho|$, we have $\rho, i \models \phi$ iff $\texttt{maximize}(\rho), i \models \mathfrak{t}(\phi)$. Now, we prove this property for formulae of size k. We consider hence a run ρ of S and $i \leq |\rho|$.
  - *Case* $\mathtt{X}\psi$*:* First, suppose that $\rho, i \models \mathtt{X}\psi$. By definition of $\models$, $i < |\rho|$ and $\rho, i{+}1 \models \psi$. By the induction hypothesis, $\texttt{maximize}(\rho), i{+}1 \models \mathfrak{t}(\psi)$. Since $i < |\rho|$, $\texttt{maximize}(\rho)[i{+}1] = \rho[i{+}1]$ and since $\texttt{maximize}(\rho)[i{+}1] \models \neg\mathtt{p}_\perp$, we get that $\texttt{maximize}(\rho), i \models \mathtt{X}(\neg\mathtt{p}_\perp \wedge \mathfrak{t}(\psi))$. Assume $\texttt{maximize}(\rho), i \models \mathtt{X}(\neg\mathtt{p}_\perp \wedge \mathfrak{t}(\psi))$. So, $\texttt{maximize}(\rho), i+1 \models \mathfrak{t}(\psi)$ and $\texttt{maximize}(\rho), i+1 \models \neg\mathtt{p}_\perp$. By construction, for all the positions j in $\texttt{maximize}(\rho)$ such that

$\mathtt{maximize}(\rho), j \models \neg \mathtt{p}_\perp$, $\mathtt{maximize}(\rho)[j]$ is a configuration in ρ. Thus, $|\rho| > i$. By the induction hypothesis, $\mathtt{maximize}^{-1}(\rho), i{+}1 \models \psi$. Therefore $\rho, i \models \mathtt{X}\ \psi$.

- *Case* $\psi \mathtt{U} \psi'$*:* First suppose that $\rho, i \models \psi \mathtt{U} \psi'$. There is $i \leq j \leq |\rho|$ such that $\rho, j \models \psi'$ and for all $k \in [i, j-1]$, we have $\rho, k \models \psi$. Since $\mathtt{maximize}(\rho)[j] = \rho[j]$, we get that $\mathtt{maximize}(\rho), j \models \neg \mathtt{p}_\perp$. Using furthermore the induction hypothesis, we deduce $\mathtt{maximize}(\rho), i \models \mathfrak{t}(\psi) \mathtt{U} (\neg \mathtt{p}_\perp \wedge \mathfrak{t}(\psi'))$.
  Now suppose that $\mathtt{maximize}(\rho), i \models \mathfrak{t}(\psi \mathtt{U} \psi')$. By definition of $\mathfrak{t}(\cdot)$, there is $i \leq j'$ such that $\mathtt{maximize}(\rho), j' \models \mathfrak{t}(\psi')$ and $\mathtt{maximize}(\rho), j' \models \neg \mathtt{p}_\perp$ and for all $k' \in [i, j'-1]$, we have $\mathtt{maximize}(\rho), k' \models \mathfrak{t}(\psi)$. Since $\mathtt{maximize}(\rho), j' \models \neg \mathtt{p}_\perp$, we know that $j' \leq |\rho|$ and $\rho[k] = \mathtt{maximize}(\rho)[k]$ for all $k \in [i, j]$. By the induction hypothesis, $\rho, j' \models \psi'$ and for all $k' \in [i, j'-1]$, we have $\rho, k' \models \psi$. Hence, $\rho, i \models \psi \mathtt{U} \psi'$.
- *Case* $\mathtt{X}^{-1}\psi$*:* First, suppose that $\rho, i \models \mathtt{X}^{-1}\psi$. By definition of $\models$, $0 < i$ and $\rho, i-1 \models \psi$. Since $i \leq |\rho|$, by the induction hypothesis, $\mathtt{maximize}(\rho), i-1 \models \mathfrak{t}(\psi)$ and consequently we get that $\mathtt{maximize}(\rho), i \models \mathtt{X}^{-1}\mathfrak{t}(\psi)$. Assume $\mathtt{maximize}(\rho), i \models \mathtt{X}^{-1}\mathfrak{t}(\psi)$. By a similar reasoning, we can prove that $\rho, i \models \mathtt{X}^{-1}\psi$.
- *Case* $\psi \mathtt{S} \psi'$*:* First suppose that $\rho, i \models \psi \mathtt{S} \psi'$. There is $0 \leq j \leq i$ such that $\rho, j \models \psi'$ and for all $k \in [j+1, i]$, we have $\rho, k \models \psi$. Using the induction hypothesis, since $i \leq |\rho|$ and hence $j \leq |\rho|$, we obtain directly that $\mathtt{maximize}(\rho), i \models \mathfrak{t}(\psi)\ \mathtt{S}\ \mathfrak{t}(\psi')$.
  Similarly if we assume $\mathtt{maximize}(\rho), i \models \mathfrak{t}(\psi \mathtt{S} \psi')$, using the fact that $i \leq |\rho|$ and the induction hypothesis we deduce that $\rho, i \models \psi \mathtt{S} \psi'$.
- *Case* $\mathtt{E}\psi$*:* First, suppose that $\rho, i \models \mathtt{E}\psi$. By definition of $\models$, there is a maximal run ρ' such that $\rho'_{\leq i} = \rho_{\leq i}$ and $\rho', i \models \psi$. By the induction hypothesis, $\mathtt{maximize}(\rho'), i \models \mathfrak{t}(\psi)$. Since by construction we have $\mathtt{maximize}(\rho')_{\leq i} = \rho'_{\leq i} = \rho_{\leq i} = \mathtt{maximize}(\rho)_{\leq i}$, we deduce

  $$\mathtt{maximize}(\rho), i \models \mathtt{E}\ \mathfrak{t}(\psi).$$

  Assume that $\mathtt{maximize}(\rho), i \models \mathtt{E}\ \mathfrak{t}(\psi)$. So, there is a run ρ' such that $\rho'_{\leq i} = \mathtt{maximize}(\rho)_{\leq i}$ and $\rho', i \models \mathfrak{t}(\psi)$. By the induction hypothesis, $\mathtt{maximize}^{-1}(\rho'), i \models \psi$. By construction we have $\mathtt{maximize}^{-1}(\rho')_{\leq i} = \rho'_{\leq i} = \mathtt{maximize}(\rho)_{\leq i} = \rho_{\leq i}$, we deduce $\rho, i \models \mathtt{E}\ \psi$.
- *Case* $\mathtt{EF}\psi$*:* The proof is the same as the one for the case $\psi \mathtt{U} \psi'$.

□

B Proof of Lemma 6

Proof.

1. By definition, an iterated path schema is a pair $\langle P, \mathtt{m} \rangle$ such that P is a minimal path schema and $\mathtt{m} \in \mathbb{N}^{\mathrm{size}(P)}$. Now assume that we have vectors $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \in \mathbb{N}^K$ such that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}$ satisfy all the conditions $\mathfrak{C}$. We perform

the construction of $P_{\mathtt{v}_t,\mathtt{v}_p}$. By definition of $P_{\mathtt{v}_t,\mathtt{v}_p}$ and $\mathfrak{C}.1$ through $\mathfrak{C}.4$ ensure that $P_{\mathtt{v}_t,\mathtt{v}_p}$ is a sequence of loops and edges from S. $\mathfrak{C}.5$ ensures that $P_{\mathtt{v}_t,\mathtt{v}_p}$ ends with a loop. $\mathfrak{C}.6$ and $\mathfrak{C}.7$ ensure that $P_{\mathtt{v}_t,\mathtt{v}_p}$ is a minimal path schema where none of the edges or loops are repeated. By construction $\mathtt{v}_p[1]$ is exactly equal to $\text{size}(P_{\mathtt{v}_t,\mathtt{v}_p})$. Thus, $\mathtt{m}_{\mathtt{v}_{it}}$ is defined to be in $\mathbb{N}^{\text{size}(P_{\mathtt{v}_t,\mathtt{v}_p})}$. $\mathfrak{C}.8$ on $\mathtt{v}_{it}$ allows us to conclude that $\langle P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}\rangle$ is an iterated path schema.
For the other side, given an iterated path schema $\langle P, \mathtt{m}\rangle$, we construct $\mathtt{v}_t$ and $\mathtt{v}_p$ as:
- $\mathtt{v}_t[1] = \mathtt{v}_p[1] = \text{size}(P)$.
- $\mathtt{v}_p[i] = j$ and $\mathtt{v}_t[i] = 0$ if $P[i-1] = \delta_j$ for some j, for all $i \in [2, \text{size}(P)+1]$.
- $\mathtt{v}_p[i] = j$ and $\mathtt{v}_t[i] = 1$ if $P[i-1] = l_j$ for some j, for all $i \in [2, \text{size}(P)+1]$.
- $\mathtt{v}_p[i] = \mathtt{v}_t[i] = 0$ for all $i \in [\text{size}(P) + 2, K]$.

Note that the above construction follows exactly the reverse steps of the construction of $P_{\mathtt{v}_t,\mathtt{v}_p}$. In other words, given an iterated path schema $\langle P, \mathtt{m}\rangle$, the above construction gives us the vectors $\mathtt{v}_t$ and $\mathtt{v}_p$ such that $P = P_{\mathtt{v}_t,\mathtt{v}_p}$. Moreover, we can construct $\mathtt{v}_{it}$ from $\mathtt{m} \in \mathbb{N}^{\text{size}(P)}$ as :
- $\mathtt{v}_{it}[i] = \mathtt{m}[i]$ for all $i \in [1, \text{size}(P)]$,
- $\mathtt{v}_{it}[i] = 0$ for all $i \in [\text{size}(P) + 1, K]$.

Clearly by the construction of $\mathtt{v}_t, \mathtt{v}_p$ and the fact that $\text{size}(P) > 0$ we can conclude that $\mathfrak{C}.1$ is satisfied. Similarly, by construction, and the fact that P is a sequence of edges and loops, $\mathfrak{C}.2$ to $\mathfrak{C}.5$ are satisfied. Due to the fact that P is a minimal path schema, $\mathfrak{C}.6$ and $\mathfrak{C}.7$ are also satisfied. Since, $\langle P, \mathtt{m}\rangle$ is an iterated path schema, $\mathfrak{C}.8$ is satisfied.

2. To construct the formula $\mathit{Schema}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it})$, we start by defining a formula ψ_i for each condition $\mathfrak{C}.i$.

$$\psi_1 \stackrel{\text{def}}{=} (\mathsf{z}_t^1 = \mathsf{z}_p^1 \wedge \mathsf{z}_p^1 = \mathsf{z}_{it}^1 \wedge \mathsf{z}_p^1 > 0 \wedge \mathsf{z}_p^1 < K) \wedge \bigwedge_{i=2}^{K} \mathsf{z}_p^1 + 1 < i \Rightarrow (\mathsf{z}_t^i = \mathsf{z}_p^i \wedge \mathsf{z}_p^i = \mathsf{z}_{it}^i \wedge \mathsf{z}_{it}^i = 0)$$

$$\psi_2 \stackrel{\text{def}}{=} \bigwedge_{i=2}^{K} (\mathsf{z}_t^i \leq 1)$$

$$\psi_3 \stackrel{\text{def}}{=} \bigwedge_{i=2}^{K} i \leq \mathsf{z}_p^1 + 1 \Rightarrow (\mathsf{z}_t^i = 0 \Rightarrow \mathsf{z}_p^i \leq \beta \wedge \mathsf{z}_p^i \geq 1)$$

$$\psi_4 \stackrel{\text{def}}{=} \bigwedge_{i=2}^{K} i \leq \mathsf{z}_p^1 + 1 \Rightarrow (\mathsf{z}_t^i = 1 \Rightarrow \mathsf{z}_p^i \leq \gamma \wedge \mathsf{z}_p^i \geq 1)$$

$$\psi_5 \stackrel{\text{def}}{=} \bigwedge_{i=2}^{K} i = \mathsf{z}_p^1 + 1 \Rightarrow \mathsf{z}_t^i = 1$$

$$\psi_6 \stackrel{\text{def}}{=} \bigwedge_{i=2}^{K} (i \leq \mathsf{z}_p^1 + 1 \Rightarrow (\bigwedge_{j=2, j\neq i}^{K} j \leq \mathsf{z}_p^1 + 1 \Rightarrow (\mathsf{z}_p^i \neq \mathsf{z}_p^j \vee \mathsf{z}_t^i \neq \mathsf{z}_t^j)))$$

$$\psi_7 \stackrel{\text{def}}{=} \bigwedge_{i=2}^{K} i \leq \mathsf{z}_p^1 \Rightarrow \bigvee_{\{(a,b,a',b') | target(\tau(a,b))=source(\tau(a'.b'))\}} (\mathsf{z}_t^i = a \wedge \mathsf{z}_p^i = b \wedge \mathsf{z}_t^{i+1} = a' \wedge \mathsf{z}_p^{i+1} = b')$$

$$\psi_8 \stackrel{\text{def}}{=} \bigwedge_{i=2}^{K} i \leq \mathsf{z}_p^1 + 1 \Rightarrow (\mathsf{z}_{it}^i > 0 \wedge (\mathsf{z}_t^i = 0 \Rightarrow \mathsf{z}_{it}^i = 1))$$

We construct the formula $Schema(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it})$ as $(\psi_1 \wedge \cdots \wedge \psi_8)$. For all $i \in [1, 8]$, $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}$ satisfies $\mathfrak{C}.i$ iff $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \models \psi_i$. For example, consider $\mathfrak{C}.6$, which expresses the minimality constraint for the minimal path schemas, stating that for any two distinct positions either the values are different or they are of different types. More formally, for all $i, j \in [1, \mathtt{v}_p[1] + 1]$, $i \neq j \Rightarrow (\mathtt{v}_p[i] \neq \mathtt{v}_t[j] \vee \mathtt{v}_t[i] \neq \mathtt{v}_t[j])$. Note that ψ_6 has a nested conjunction allowing to compare every possible pair of positions $\langle i, j \rangle$. Moreover, for every possible pair, we check exactly the condition stated in $\mathfrak{C}.6$. Clearly, $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}$ satisfies $\mathfrak{C}.6$ iff $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \models \psi_6$. Finally, the formula *Schema* states that all formulae ψ_i hold for the vectors $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}$. Hence it is easy to see that, $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \models Schema(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it})$ iff $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}$ satisfies $\mathfrak{C}$.

□

C Proof of Lemma 7

Proof. Let $\mathtt{w}_0, \ldots, \mathtt{w}_{\beta+\gamma-1}, \mathtt{c}_0 \in \mathbb{N}^{n+1}$ and $\mathfrak{p}_0, \ldots \mathfrak{p}_{\beta+\gamma-1} \in \mathbb{N}$ and $\mathtt{v}_t$, $\mathtt{v}_p$, $\mathtt{v}_{it}$ in $\mathbb{N}^K$ such that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \models Schema(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it})$. By Lemma 6, we know that $\mathtt{v}_t$, $\mathtt{v}_p$ and $\mathtt{v}_{it}$ respect the conditions in $\mathfrak{C}$ and consequently that $\langle P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}} \rangle$ is an iterated path schema.

First we suppose that $\mathtt{w}_0, \ldots, \mathtt{w}_{\beta+\gamma-1}, \mathfrak{p}_0, \ldots, \mathfrak{p}_{\beta+\gamma-1}, \mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0 \models Witness$.

- The subformula

$$\bigvee_{s=0}^{1} \bigvee_{j=1}^{\max(\beta,\gamma)} \mathsf{z}_p^2 = j \wedge \mathsf{z}_t^2 = s \wedge \mathsf{x}_0^1 = source(\tau(s, j))$$

  ensures that $q_0 = source(P_{\mathtt{v}_t,\mathtt{v}_p}[1])$; in fact it states that $\mathtt{c}_0[1] = source(\tau(\mathtt{v}_t[2], \mathtt{v}_p[2]))$. Consequently, we can define the pseudo-run $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)$.
- Let $N = \text{size}(P_{\mathtt{v}_t,\mathtt{v}_p})$. We recall that $N = \mathtt{v}_t[1]$. We will now show that $\mathtt{w}_j$ is the jth witness configuration of $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)$ and $\mathfrak{p}_j$ is its position for all $j < N$.
  - First thanks to the part of formula saying $\mathtt{d}_0 = 0 \wedge \mathsf{X}_0 = \mathsf{W}_0$, we know that $\mathfrak{p}_0 = 0$ and that $\mathtt{w}_0 = \mathtt{c}_0$, whence $\mathtt{w}_0$ is effectively the witness configuration at position 0. Now thanks to the part of the formula saying that $\bigwedge_{i=1}^{\beta+\gamma-1} (i < \mathsf{z}_t^1 \Rightarrow \bigvee_{s=0}^{1} \bigvee_{j=1}^{\max(\beta,\gamma)} (\mathsf{z}_p^{i+1} = j \wedge \mathsf{z}_t^{i+1} = s \wedge \mathtt{d}_i = \mathtt{d}_{i-1} + \mathsf{z}_{it}^{i+1} *$

$length(\tau(s,j))))$, we have that for all $1 < j < N$,

$$\mathtt{p}_j = \sum_{k=1}^{k=j} \mathtt{v}_{it}[k+1] * length(\tau(\mathtt{v}_t[k+1], \mathtt{v}_p[k+1]))$$

and this allows us to deduce that $\mathtt{p}_j$ is the position of the jth witness configuration (using the definition of witness configuration).

- Similarly, the part of the formula saying $\bigwedge_{i=1}^{\beta+\gamma-1} (i < \mathsf{z}_t^1 \Rightarrow \bigvee_{s=0}^{1} \bigvee_{j=1}^{\max(\beta,\gamma)} (\mathsf{z}_p^{i+1} = j \wedge \mathsf{z}_t^{i+1} = s \wedge \mathsf{w}_i^1 = target(\tau(t,j)) \wedge \mathsf{W}_i = \mathsf{W}_{i-1} + \mathsf{z}_{it}^{i+1} * update(\tau(s,i))))$ ensures us that for all $j < N$, $\mathtt{w}_j[1] = target(\tau(\mathtt{v}_t[j+1], \mathtt{v}_p[j+1]))$ and for all $i \in [2, n+1]$,

$$\mathtt{w}_j[i] = \mathtt{c}_0[i] + \sum_{k=1}^{k=j} \mathtt{v}_{it}[k+1] * update(\tau(\mathtt{v}_t[k+1], \mathtt{v}_p[k+1]))[i]$$

and consequently $\mathtt{w}_j$ corresponds effectively to the jth witness configuration. Furthermore since for all $j < N$, $\mathtt{w}_j$ belongs to $\mathbb{N}^{n+1}$, then we deduce that the pseudo-run $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)$ is positive.

We now assume that $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)$ is well-defined and positive and for all $j \in [0, \beta+\gamma-1]$, if $j < \text{size}(P_{\mathtt{v}_t,\mathtt{v}_p})$, then $\mathtt{w}_j$ is the jth witness configuration of $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)$ and $\mathtt{p}_j$ its position. Let $N = \text{size}(P_{\mathtt{v}_t,\mathtt{v}_p})$. We recall that $N = \mathtt{v}_t[1]$ by definition of $P_{\mathtt{v}_t,\mathtt{v}_p}$. By definition of the witness configurations, the following assertions hold:

- $\mathtt{p}_0 = 0$ and $\mathtt{w}_0 = \mathtt{c}_0$ and $\mathtt{c}_0[1] = source(P_{\mathtt{v}_t,\mathtt{v}_p}[1])$, whence $\mathtt{w}_0[1] = source(P_{\mathtt{v}_t,\mathtt{v}_p}[1])$;
- For all $1 < j < N$,

$$\begin{aligned}\mathtt{p}_j &= \textstyle\sum_{k=1}^{k=j} \mathtt{v}_{it}[k+1] * length(P_{\mathtt{v}_t,\mathtt{v}_p}[k]) \\ &= \textstyle\sum_{k=1}^{k=j} \mathtt{v}_{it}[k+1] * length(\tau(\mathtt{v}_t[k+1], \mathtt{v}_p[k+1])) \\ &= \mathtt{p}_{j-1} + \mathtt{v}_{it}[j+1] * length(\tau(\mathtt{v}_t[j+1], \mathtt{v}_p[j+1])).\end{aligned}$$

- For all $1 < j < N$,

$$\mathtt{w}_j[1] = target(P_{\mathtt{v}_t,\mathtt{v}_p}[j]) = target(\tau(\mathtt{v}_t[j+1], \mathtt{v}_p[j+1])).$$

- For all $1 < j < N$, for all $i \in [2, n+1]$,

$$\begin{aligned}\mathtt{w}_j[i] &= \textstyle\sum_{k=1}^{k=j} \mathtt{v}_{it}[k+1] * update(P_{\mathtt{v}_t,\mathtt{v}_p}[k])[i] \\ &= \textstyle\sum_{k=1}^{k=j} \mathtt{v}_{it}[k+1] * update(\tau(\mathtt{v}_t[k+1], \mathtt{v}_p[k+1]))[i] \\ &= \mathtt{w}_{j-1}[i] + \mathtt{v}_{it}[j+1] * update(\tau(\mathtt{v}_t[j+1], \mathtt{v}_p[j+1]))[i].\end{aligned}$$

These assertions allow us to conclude that

$$\mathtt{w}_0, \ldots, \mathtt{w}_{\beta+\gamma-1}, \mathtt{p}_0, \ldots, \mathtt{p}_{\beta+\gamma-1}, \mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0 \models Witness.$$

□

D Proof of Lemma 8

Proof. **Proof of (1.).** We assume that there exist $\mathtt{c}_0, \mathtt{c} \in \mathbb{N}^{n+1}$ and $i \in \mathbb{N}$ and $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \in \mathbb{N}^K$ such that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i, \mathtt{c} \models \mathit{Conf}$. Thanks to the conjunct $\mathit{Schema}(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it})$, we have that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \models \mathit{Schema}$ and using Lemma 6 we know that $\langle P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}} \rangle$ is an iterated path schema. Furthermore we know that there exist $\mathtt{w}_0, \ldots, \mathtt{w}_{\beta+\gamma-1} \in \mathbb{N}^{n+1}$ and $\mathfrak{p}_0, \ldots, \mathfrak{p}_{\beta+\gamma-1} \in \mathbb{N}$ such that

$$\mathtt{w}_0, \ldots, \mathtt{w}_{\beta+\gamma-1}, \mathfrak{p}_0, \ldots, \mathfrak{p}_{\beta+\gamma-1}, \mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0 \models \mathit{Witness},$$

hence thanks to Lemma 7, we have that $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)$ is well-defined and positive and for all $j \in [0, \beta+\gamma-1]$, if $j < \text{size}(P_{\mathtt{v}_t,\mathtt{v}_p})$, then $\mathtt{w}_j$ represents the jth witness configuration of $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)$ and $\mathfrak{p}_j$ its position. Now, we proceed on a case analysis on the value of i:

- First assume that there exists $j \leq \mathtt{v}_p[1] = \text{size}(P_{\mathtt{v}_t,\mathtt{v}_p})$ such that $i = \mathfrak{p}_j$ and $\mathtt{v}_t[j+2] = 0$. We know from the formula that $\mathtt{c} = \mathtt{w}_j$ and hence since $\mathtt{w}_j = \rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, c_0)[\mathfrak{p}_j]$ (by definition of witness configuration), we deduce that $\mathtt{c} = \rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, c_0)[i]$. Furthermore we have that $\mathtt{v}_t[j+2] = 0$, hence $P_{\mathtt{v}_t,\mathtt{v}_p}[j+1] \in \Delta$ and we also have that $\mathtt{c}[2], \ldots, \mathtt{c}[n+1] \models \mathit{guard}(\delta_{\mathtt{v}_p[j+2]})$. Using the definition of iterated path schema we have that

$$\begin{aligned}\delta_{\mathtt{v}_p[j+2]} &= P_{\mathtt{v}_t,\mathtt{v}_p}[j+1] \\ &= \mathit{trans}(\langle P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}} \rangle)[\sum_{k=1}^{k=j} \mathit{length}(P_{\mathtt{v}_t,\mathtt{v}_p}[k]) * \mathtt{v}_{it}[k+1] + 1]\end{aligned}$$

  and using the definition of the position of the jth witness configuration, we deduce that $\delta_{\mathtt{v}_p[j+2]} = \mathit{trans}(\langle P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}} \rangle)[\mathfrak{p}_j + 1] = \mathit{trans}(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}})[i+1]$. Consequently $\mathtt{c}[2], \ldots, \mathtt{c}[n+1] \models \mathit{guard}(\mathit{trans}(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}})[i+1])$.
- Now assume that there exists $j \leq \mathtt{v}_p[1] = \text{size}(P_{\mathtt{v}_t,\mathtt{v}_p})$ such that $\mathfrak{p}_j \leq j < \mathfrak{p}_{j+1}$ and $\mathtt{v}_t[j+2] = 1$ (hence $P_{\mathtt{v}_t,\mathtt{v}_p}[j+1]$ is a loop). Then, we have $\mathtt{w}_j, i - \mathfrak{p}_j, \mathtt{c} \models \mathit{ConfLoop}_{l_k}$ with $k = \mathtt{v}_p[j+2]$. Then let $u \in [0, \mathit{length}(l_k)-1]$ and $v \in \mathbb{N}$ such that $i - \mathfrak{p}_j = v * \mathit{length}(l_k) + u$. Note that since $\mathfrak{p}_j \leq i < \mathfrak{p}_{j+1}$, we know that the pseudo-run $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, c_0)$ at position i lies exactly in the loop l_k and that it has taken this loop v times exactly and the u first transitions. Moreover, the next transition will still be one of the loops (otherwise we would have $i \geq \mathfrak{p}_{j+1}$). The formula enforces that $\mathtt{c}[1] = \mathit{source}(l_k[u+1])$ and for all $2 \leq f \leq n+1$, we have $\mathtt{c}[f] = \mathtt{w}_j[f] + \sum_{r=1}^{r=u} \mathit{update}(l_k[r])[f]$, hence this allows us to deduce that $\mathtt{c} = \rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, c_0)[i]$. Furthermore we also have that $\mathtt{c}[2], \ldots, \mathtt{c}[n+1] \models \mathit{guard}(l_k[u+1])$, which allows us to deduce that $\mathtt{c}[2], \ldots, \mathtt{c}[n+1] \models \mathit{guard}(\mathit{trans}(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}})[i+1])$. In fact, we have $\mathit{trans}(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}})[i+1] = \mathit{trans}(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}})[\mathfrak{p}_j + (i - \mathfrak{p}_j) + 1] = \mathit{trans}(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}})[\mathfrak{p}_j + v * \mathit{length}(l_k) + u + 1] = l_k[u+1]$.
- The case where there exist $j = \mathtt{v}_p[1] = \text{size}(P_{\mathtt{v}_t,\mathtt{v}_p})$ and $\mathfrak{p}_j \leq j$ is similar to the previous one, the only difference being that this time the position of the run is in the last loop of the minimal path schema $P_{\mathtt{v}_t,\mathtt{v}_p}$.

The other direction which consists in proving that if $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \models \mathit{Schema}$ and $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, \mathtt{c}_0)$ is well-defined and $\mathtt{c} = \rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, c_0)[i]$ and $\mathtt{c}[2], \cdots, \mathtt{c}[n+1] \models \mathit{guard}(\mathit{trans}(\langle P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}\rangle)[i+1])$ then $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i, \mathtt{c} \models \mathit{Conf}$ can be done in a similar way.

Proof of (2.). Let us check that the formula *Run* is what we are looking for. First, we assume that there exist $\mathtt{c}_0 \in \mathbb{N}^{n+1}$ and $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \in \mathbb{N}^{\beta+\gamma+1}$ such that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0 \models \mathit{Run}$, then for all $i \in \mathbb{N}$, there exists $\mathtt{c} \in \mathbb{N}^{n+1}$ such that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i, \mathtt{c} \models \mathit{Conf}$. Thanks to the previous property, we have $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \models \mathit{Schema}$ and $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, c_0)$ is well-defined and positive and $\mathtt{c} = \rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, c_0)[i]$ is represented by $\mathtt{c}$ and

$$\mathtt{c}[2], \cdots, \mathtt{c}[n+1] \models \mathit{guard}(\mathit{trans}(\langle P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}\rangle)[i+1]).$$

Consequently the pseudo-run $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, c_0)$ is a run.

Now assume that there exist $\mathtt{c}_0 \in \mathbb{N}^{n+1}$ and $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \in \mathbb{N}^{\beta+\gamma+1}$ such that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it} \models \mathit{Schema}$ and $\rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, c_0)$ is well-defined and is a run. Let $i \in \mathbb{N}$ and let $\mathtt{c} = \rho(P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}, c_0)[i]$, by definition of a run we have $\mathtt{c}[2], \cdots, \mathtt{c}[n+1] \models \mathit{guard}(\mathit{trans}(\langle P_{\mathtt{v}_t,\mathtt{v}_p}, \mathtt{m}_{\mathtt{v}_{it}}\rangle)[i+1])$, consequently we deduce that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0, i, \mathtt{c} \models \mathit{Conf}$. This allows us to deduce that $\mathtt{v}_t, \mathtt{v}_p, \mathtt{v}_{it}, \mathtt{c}_0 \models \mathit{Run}$.

All the formulae involved in the construction use generalised conjunctions/disjunctions of length linear in the sum of the following values: number of states α, number of transitions β, number of simple loops γ, number of counters n. So, it is easy to check that we only need logarithmic-space to build the formulae.