Flat counter automata almost everywhere!*

Jérôme Leroux[1] and Grégoire Sutre[2]

[1] IRISA, Vertecs Project, Campus de Beaulieu, Rennes, France
jleroux@irisa.fr
[2] LaBRI, CNRS UMR 5800, Domaine Universitaire, Talence, France
sutre@labri.fr

Abstract. This paper argues that flatness appears as a central notion in the verification of counter automata. A counter automaton is called flat when its control graph can be "replaced", equivalently w.r.t. reachability, by another one with no nested loops. From a practical view point, we show that flatness is a necessary and sufficient condition for termination of accelerated symbolic model checking, a generic semi-algorithmic technique implemented in successful tools like FAST, LASH or TREX. From a theoretical view point, we prove that many known semilinear subclasses of counter automata are flat: reversal bounded counter machines, lossy vector addition systems with states, reversible Petri nets, persistent and conflict-free Petri nets, etc. Hence, for these subclasses, the semilinear reachability set can be computed using a *uniform* accelerated symbolic procedure (whereas previous algorithms were specifically designed for each subclass).

1 Introduction

Petri nets and *counter automata* are widely used formalisms to model concurrent distributed systems. Basically, a counter automaton is a finite-state automaton extended with counters that hold nonnegative integer values. Operations on counters can be defined by formulas in Presburger arithmetic. As the counters are unbounded, counter automata are naturally *infinite-state* systems.

Various formalisms have been proposed to model desired properties on systems. In this work, we only consider *safety* properties: these properties (of the original system) may often be expressed by *reachability properties* on the model.

Reachability properties are algorithmically checkable for *finite-state* systems (and efficient implementations exist). However, the situation is more complex for *infinite-state* systems: the reachability problem is undecidable even for restricted classes of systems, such as Minsky machines [Min67].

Dedicated algorithms for counter automata. Many specialized algorithms have been designed to solve verification problems for various classes of counter automata. The reachability problem for Petri nets has been proved decidable [May84, Kos82]. The binary reachability relation is effectively semilinear for reversible Petri nets [Tai68]

* This work was supported by the French Ministry of Research (Project PERSÉE of the ACI Sécurité et Informatique).

Improved and extended (unofficial) version of the paper published in:
Proc. 3rd Int. Symp. Automated Technology for Verification and Analysis (ATVA'05), Taipei, Taiwan, Oct. 2005, volume 3707 of Lecture Notes in Computer Science, pages 489–503. Springer, 2005.

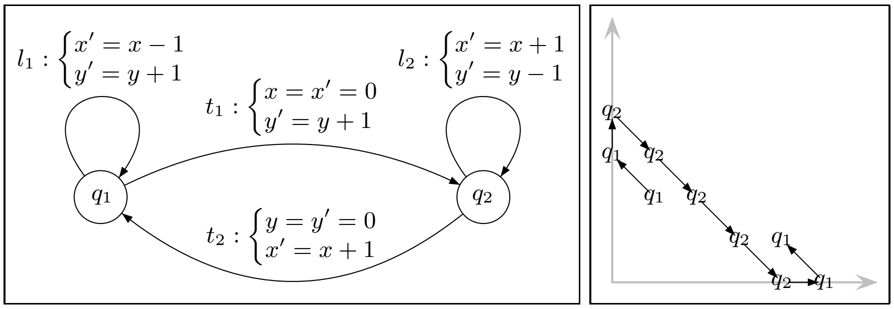

Fig. 1. A non-flat counter automaton.

and for BPP-nets [Esp97], and the reachability set post^* is effectively semilinear for cyclic Petri nets [AK77], for persistent Petri nets [LR78, May81] and for regular Petri nets [VVN81]. The reachability sets post^* and pre^* are effectively semilinear for reversal-bounded counter machines [Iba78], for lossy VASS [BM99] and for 2-dimensional VASS [HP79]. It was later shown that post^* / pre^* are still effectively semilinear for various extensions of 2-dim VASS [FS00b, FS00a]. However, these methods suffer from serious drawbacks: (1) they cannot be easily extended or combined, (2) from an implementation perspective, a dedicated tool would be needed for each specialized algorithm, and (3) in practice, counter automata rarely belong entirely to one of these semilinear classes. Thus, generic symbolic model-checking techniques for general (undecidable) classes have been recently developped and implemented.

Accelerated symbolic model-checking. Verification of reachability properties usually proceeds through an iterative fixpoint computation of the *forward reachability set* post^* (resp. *backward reachability set* pre^*), starting from the initial states (resp. from the error states). When the state space is infinite, finite *symbolic representations* for sets of states are required. To help termination of this fixpoint computation, so-called *acceleration* techniques (or *meta-transitions*) are applied [BW94, BGWW97, BH99, FIS03, FL02]. Basically, acceleration consists in computing in one step the effect of iterating a given loop (of the control flow graph). Accelerated symbolic model checkers such as LASH [Las], TREX [ABS01], and FAST [BFLP03] implement this approach.

Even though it behaves well in practice, accelerated symbolic model-checking is only a *semi*-algorithm: it does not provide any guarantee of termination. For instance, iteration of loops is not sufficient to compute the whole semilinear reachability set of the counter automata depicted in figure 1, with initial state $(q_1, (0, 0))$ (see Examples 2.4 and 4.5). Thus, we would like to combine the best of both approaches, by integrating, for each known semilinear class, the dedicated algorithm's technology into improved acceleration techniques that would ensure termination of the generic accelerated semi-algorithm for this class. A first step towards this objective consists in characterizing the classes for which the generic accelerated semi-algorithm fials to terminate.

Our contribution. In this work, we investigate termination of accelerated symbolic model-checking for known semilinear classes of counter automata. A natural notion in this framework is *flatness* [FO97, CJ98]: a counter automaton S is called *flat*[3] when its control graph can be "replaced", equivalently w.r.t. reachability, by another one with no nested loops. We show that (global) flatness is a necessary and sufficient condition for termination of (binary) reachability set computations by acceleration-based semi-algorithms. In particular, we get that accelerated symbolic model checkers terminate on a given system iff this system is flat (and a suitable search strategy is used).

We then turn our attention to the analysis of flatness for known semilinear classes of counter automata. We show that most of the known semilinear classes of counter automata (in particular the ones cited above) are flat. Our main technical contributions are the proofs of flatness for the following classes: reversal-bounded counter machines, reversible Petri nets and conflict-free Petri nets. In particular, we obtain that the binary reachability relation is effectively semilinear for conflict-free Petri nets. We also show that cyclic Petri nets, persistent Petri nets, regular Petri nets and Lossy / Inserting counter machines are flat, and we recall that BPP-nets and 2-dim VASS are flat. As flatness implies effective semilinearity of the forward / binary reachability set, our results give new "uniform" proofs that these classes are semilinear. In particular, we obtain a simpler semilinearity proofs for reversal-bounded counter machines and reversible Petri nets.

It is also remarkable that accelerated symbolic model checkers designed to analyse counter automata, such as LASH and FAST, terminate on all these classes. From a practical viewpoint, our approach has several benefits: (1) we can apply a *generic* algorithm, which was designed for a much larger class of (undecidable) systems, and (2) the — forward, backward and binary — reachability sets can be computed using the same generic algorithm.

Outline. The paper is organized as follows. Section 2 presents general counter automata. We introduce the notion of flatness in Section 3 and we show that flatness is a necessary and sufficient condition for termination of accelerated symbolic model-checking. In the last two sections, we show that many known semilinear restricted classes of counter automata are flat: Section 4 deals with classes of counter machines, and Section 5 deals with classes of Petri nets.

Proofs. For the sake of completeness, all results are proved in detail (some proofs are in appendix). This paper is the long version of our ATVA 2005 paper.

2 General Counter Automata

This section is devoted to the presentation of general counter automata. We will consider in section 4 a more effective subclass of counter automa based on guarded commands. We first give basic definitions and notations that will be used throughout the paper.

[3] Our notion of flatness is actually more general than in [CJ98]: there, a system is called flat when it contains no nested loops.

2.1 Numbers, Vectors, Relations

Let $\mathbb{Z}$ (resp. $\mathbb{N}$, $\mathbb{Z}^-$, $\mathbb{Q}$, $\mathbb{Q}_+$) denotes the set of *integers* (resp. *nonnegative integers*, *nonpositive integers*, *rational numbers*, *nonnegative rational numbers*). We denote by $\leq$ the *usual total order on* $\mathbb{Q}$. Given $k, l \in \mathbb{N}$, we write $[k \mathinner{..} l]$ (resp. $[k \mathinner{..} \infty[$) for the *interval of integers* $\{i \in \mathbb{N} \;/\; k \leq i \leq l\}$ (resp. $\{i \in \mathbb{N} \;/\; k \leq i\}$). We write $|X|$ the *cardinal* of any finite set X.

Given a set X and $n \in \mathbb{N}$, we write X^n for the set of n-dim *vectors* x of elements in X. For any index $i \in [1 \mathinner{..} n]$, we denote by $\mathsf{x}[i]$ the i^{th} *component* of an n-dim vector x.

We now focus on n-dim vectors of (integer or rational) numbers. We write $\mathsf{0}$ for the *all zero vector*: $\mathsf{0}[i] = 0$ for all $i \in [1 \mathinner{..} n]$. We also denote by $\leq$ the *usual partial order on* $\mathbb{Q}^n$, defined by $\mathsf{x} \leq \mathsf{y}$ if for all $i \in [1 \mathinner{..} n]$ we have $\mathsf{x}[i] \leq \mathsf{y}[i]$.

Operations on n-dim vectors are componentwise extensions of their scalar counterpart (e.g. for $\mathsf{x}, \mathsf{x}' \in \mathbb{Q}^n$, $\mathsf{x} + \mathsf{x}'$ is the vector $\mathsf{y} \in \mathbb{Q}^n$ defined by $\mathsf{y}[i] = \mathsf{x}[i] + \mathsf{x}'[i]$ for all $i \in [1 \mathinner{..} n]$). For $\alpha \in \mathbb{Q}$ and $\mathsf{x} \in \mathbb{Q}^n$, $\alpha\,\mathsf{x}$ is the vector $\mathsf{y} \in \mathbb{Q}^n$ defined by $\mathsf{y}[i] = \alpha\,\mathsf{x}[i]$ for all $i \in [1 \mathinner{..} n]$.

These operations are classically extended on sets of n-dim vectors (e.g. for $P, P' \subseteq \mathbb{Q}^n$, $P + P' = \{\mathsf{p} + \mathsf{p}' \;/\; \mathsf{p} \in P, \mathsf{p}' \in P'\}$). Moreover, in an operation involving sets of n-dim vectors, we shortly write x for the singleton $\{\mathsf{x}\}$ (e.g. for $P \subseteq \mathbb{Q}^n$ and $\mathsf{x} \in \mathbb{Q}^n$, we write $\mathsf{x} + P$ for $\{\mathsf{x}\} + P$).

A *binary relation* R on some set X is any subset of $X \times X$. We shortly write $x \, R \, x'$ whenever $(x, x') \in R$. Given a set Y, we denote by $R[Y]$ the *relational image* of Y by R, defined by $R[Y] = \{x \in X \;/\; \exists y \in Y, y \, R \, x\}$. The *inverse* of a binary relation R on X is the binary relation R^{-1} on X defined by $x \, R^{-1} \, x'$ iff $x' \, R \, x$. We say R is *symmetric* if $R = R^{-1}$. Given two binary relations R_1, R_2 on X, the *composed binary relation* $R_1 \cdot R_2$ on X is defined by $x \, (R_1 \cdot R_2) \, x'$ if we have $x \, R_1 \, y$ and $y \, R_2 \, x'$ for some $y \in X$. We denote by R^* the *reflexive and transitive closure* of R. The *identity relation on* X is the binary relation $Id_X = \{(x, x) \;/\; x \in X\}$. In the rest of the paper, we will only consider binary relations, and they will shortly be called *relations*.

2.2 Presburger Arithmetic and Semilinear Sets

Presburger arithmetic (the first order additive theory over the integers $\langle \mathbb{Z}, +, \leq \rangle$) is a decidable logic used in a large range of applications. As described in [Lat04], this logic is central in many areas including integer programming problems, compiler optimization techniques, program analysis tools and model-checking.

Presburger-definable subsets of $\mathbb{Z}^n$ may also be represented in terms of *semilinear sets* [GS66]. For any subset $P \subseteq \mathbb{Z}^n$, we denote by P^* the set of all (finite) linear combinations of vectors in P:

$$P^* = \left\{ \sum_{i=0}^{k} c_i \, \mathsf{p}_i \;/\; k, c_0, \ldots, c_k \in \mathbb{N} \text{ and } \mathsf{p}_0, \ldots, \mathsf{p}_k \in P \right\}$$

A subset $S \subseteq \mathbb{Z}^n$ is said to be a *linear set* if $S = (\mathsf{x} + P^*)$ for some $\mathsf{x} \in \mathbb{Z}^n$ and for some finite subset $P \subseteq \mathbb{Z}^n$; moreover x is called the *basis* and vectors in P are called *periods*. A *semilinear set* is any finite union of linear sets. Let us recall that semilinear sets are precisely the subsets of $\mathbb{Z}^n$ that are definable in Presburger arithmetic [GS66].

Observe that any finite non empty set Q can be "encoded" using a bijection η from Q to $[1 \,..\, |Q|]$. Thus, these semilinearity notions and Presburger-definability notions naturally carry[4] over subsets of $Q \times \mathbb{Z}^n$ and over relations on $Q \times \mathbb{Z}^n$.

2.3 Counter Automata

Definition 2.1. *A* n-dim counter automaton $\mathbb{S}$ *(*counter automaton *for short), is defined as a tuple* $\mathbb{S} = (Q, T, \alpha, \beta, (G_t)_{t \in T})$*, where* Q *is a finite non empty set of* locations*,* T *is a finite non empty set of* transitions*,* $\alpha : T \to Q$ *and* $\beta : T \to Q$ *are the* source *and* target *mappings, and* $(G_t)_{t \in T}$ *is a family of binary relations on* $\mathbb{N}^n$ *called* flow guards*.*

An n-dim counter automaton is basically a finite graph whose edges are labeled by relations over n-dim vector of integers. Each component $i \in [1 \,..\, n]$ corresponds to a counter ranging over $\mathbb{N}$. Operationally, control flows from one location to another along transitions, and counters simultaneously change values according to the transition's flow guard.

Formally, let $\mathbb{S} = (Q, T, \alpha, \beta, (G_t)_{t \in T})$ be a n-dim counter automaton. The *set of configuration* $\mathcal{C}_{\mathbb{S}}$ of $\mathbb{S}$ is $Q \times \mathbb{N}^n$, and the semantics of each transition $t \in T$ is given by the *action reachability relation* $\mathcal{R}_{\mathbb{S}}(t)$ over $\mathcal{C}_{\mathbb{S}}$ defined by:

$$(q, \mathsf{x}) \ \mathcal{R}_{\mathbb{S}}(t) \ (q', \mathsf{x}') \qquad \text{if} \qquad q = \alpha(t) \ \text{ and } \ q' = \beta(t) \ \text{ and } \ \mathsf{x} \ G_t \ \mathsf{x}'$$

Definition 2.2. *An* initialized n-dim counter automaton $(\mathbb{S}, I)$ *is a tuple such that* $\mathbb{S}$ *is an* n*-dim counter automaton and* $I \subseteq \mathcal{C}_{\mathbb{S}}$*.*

We write T^+ for the set of all *non empty words* $t_0 \cdots t_k$ with $t_i \in T$, and ε denotes the *empty word*. The set $T^+ \cup \{\varepsilon\}$ of all *words* π over T is denoted by T^*. For any word $\pi \in T^*$ and for any $t \in T$, we let $|\pi|_t$ denote the number of occurences of t in π. Flow guards and transition reachability relations are naturally extended to words:

$$\begin{cases} G_\varepsilon = Id_{\mathbb{N}^n} \\ G_{\pi \cdot t} = G_\pi \cdot G_t \end{cases} \qquad \begin{cases} \mathcal{R}_{\mathbb{S}}(\varepsilon) = Id_{\mathcal{C}_{\mathbb{S}}} \\ \mathcal{R}_{\mathbb{S}}(\pi \cdot t) = \mathcal{R}_{\mathbb{S}}(\pi) \cdot \mathcal{R}_{\mathbb{S}}(t) \end{cases}$$

A *language* over T is any subset L of T^*. We also extend flow guards and reachability relations to languages : $G_L = \bigcup_{\pi \in L} G_\pi$ and $\mathcal{R}_{\mathbb{S}}(L) = \bigcup_{\pi \in L} \mathcal{R}_{\mathbb{S}}(\pi)$. For any language $L \subseteq T^*$ and for any set of configurations $I \subseteq \mathcal{C}_{\mathbb{S}}$, we respectively denote by $\mathsf{post}_{\mathbb{S}}(L, I)$ and by $\mathsf{pre}_{\mathbb{S}}(L, I)$ the set of *successor* configurations $(\mathcal{R}_{\mathbb{S}}(L))[I]$ and the set of *predecessor* configurations $(\mathcal{R}_{\mathbb{S}}(L))^{-1}[I]$.

Definition 2.3. *Given a counter automaton* $\mathbb{S}$*, the* one-step reachability relation *of* $\mathbb{S}$ *is the relation* $\mathcal{R}_{\mathbb{S}}(T)$*, shortly written* $\mathcal{R}_{\mathbb{S}}$*. The* global reachability relation *of* $\mathbb{S}$ *is the relation* $\mathcal{R}_{\mathbb{S}}(T^*)$*, shortly written* $\mathcal{R}_{\mathbb{S}}^*$*. Given a subset* $I \subseteq \mathcal{C}_{\mathbb{S}}$*, the sets* $\mathsf{post}_{\mathbb{S}}(T^*, I)$*, shortly written* $\mathsf{post}_{\mathbb{S}}^*(I)$*, and* $\mathsf{pre}_{\mathbb{S}}(T^*, I)$*, shortly written* $\mathsf{pre}_{\mathbb{S}}^*(I)$*, are respectively called the* forward reachability set *of* $(\mathbb{S}, I)$ *and the* backward reachability set *of* $(\mathbb{S}, I)$*.*

[4] Obviously, the extension of these notions does not depend on the "encoding" η.

Remark that the global reachability relation is the reflexive and transitive closure of the one-step reachability relation. A *reachability subrelation* is any relation $R \subseteq \mathcal{R}_{\mathcal{S}}^*$. For the reader familiar with transition systems, the operational semantics of $\mathcal{S}$ can be viewed as the infinite-state transition system $(\mathcal{C}_{\mathcal{S}}, \mathcal{R}_{\mathcal{S}})$.

The *inverse counter automaton* $\mathcal{S}^{-1}$ of a counter automaton $\mathcal{S}$ is obtained from $\mathcal{S}$ by replacing the flow guards G_t with their inverse G_t^{-1}. As $\mathsf{pre}_{\mathcal{S}}(L, I) = \mathsf{post}_{\mathcal{S}^{-1}}(L, I)$ for every $L \subseteq T^*$ and $I \subseteq \mathcal{C}_{\mathcal{S}}$, we restrict our attention (without loss of generality) to the *global reachability relation* and the *forward reachability set* (shortly called *reachability set* from now on).

Consider two locations q and q' in a system $\mathcal{S}$. A word $\pi \in T^*$ is called a *path from q to q'* if either (1) $\pi = \varepsilon$ and $q = q'$, or (2) $\pi = t_0 \cdots t_k$ with $k \in \mathbb{N}$ and satisfies: $q = \alpha(t_0)$, $q' = \beta(t_k)$ and $\beta(t_{i-1}) = \alpha(t_i)$ for every $i \in [1 .. k]$. A path from q to q is called a *loop on q*, or shortly a *loop*. We denote by $\Pi_{\mathcal{S}}(q, q')$ the set of all paths from q to q' in $\mathcal{S}$. The set $\bigcup_{q,q' \in Q} \Pi_{\mathcal{S}}(q, q')$ of all *paths* in $\mathcal{S}$ is written $\Pi_{\mathcal{S}}$. A *trace* of an initialized counter automaton $(\mathcal{S}, I)$ is any word $\pi \in T^*$ such that $\mathsf{post}(\pi, I) \neq \emptyset$. Note that every trace is a path, but the converse is not true.

Notation. In the following, we will simply write $\mathcal{R}$ (resp. post, Π, $\mathcal{C}$) instead of $\mathcal{R}_{\mathcal{S}}$ (resp. $\mathsf{post}_{\mathcal{S}}$, $\Pi_{\mathcal{S}}, \mathcal{C}_{\mathcal{S}}$), when the underlying counter automaton is unambiguous. We will also sometimes write $\rightarrow$ (resp. $\xrightarrow{\sigma}$, $\xrightarrow{L}$, $\xrightarrow{*}$) instead of $\mathcal{R}$ (resp. $\mathcal{R}(\sigma)$, $\mathcal{R}(L)$, $\mathcal{R}^*$).

Example 2.4. Consider the 2-dim counter automaton $\mathcal{E}$ depicted in figure 1. Counters are denoted by x and y and flow guards are given by predicates over x, y, x', and y' (with an implicit conjonction between equalities). Intuitively, the loop l_1 on location q_1 transfers the contents of the first counter into the second counter, while the loop l_2 on location q_2 does the converse. *Intermediate locations* along $(q_1, (1, 2)) \xrightarrow{l_1 t_1 l_2^4 t_2 l_1} (q_1, (4, 1))$ are also depicted above. This counter automaton exhibits a simple global reachability relation, since it is readily seen that $(q_1, (x, y)) \xrightarrow{*} (q_1, (x', y'))$ if and only if: $(x'+y')-(x+y)$ is even, and $x'+y' = x+y$ implies $x' \leq x$. Relation $(q_2, (x, y)) \xrightarrow{*} (q_2, (x', y'))$ is similar, and thus we obtain, by composition with relations $\mathcal{R}_{\mathcal{E}}(t_1)$ and $\mathcal{R}_{\mathcal{E}}(t_2)$, that $\mathcal{E}$ has a semilinear global reachability relation. □

3 Flatness as a criterion for acceleration completeness

We now investigate termination of accelerated symbolic reachability computations on counter automata. An important concept used in this paper is that of *semilinear path scheme (SLPS)* [LS04].

Definition 3.1. *[LS04] A* linear path scheme *(LPS for short) for a counter automaton $\mathcal{S}$ is any language $\rho \subseteq \Pi_{\mathcal{S}}$ of the form $\rho = \sigma_0 \theta_1^* \sigma_1 \cdots \theta_k^* \sigma_k$ where $\sigma_0, \theta_1, \sigma_1, \ldots, \theta_k, \sigma_k$ are words. A* semilinear regular path scheme *(SLPS for short) is any finite union of LPS.*

Definition 3.2. *A counter automaton $\mathcal{S}$ (resp. initialized counter automaton $(\mathcal{S}, I)$) is called* globally flat *(resp.* flat*) if there exists an SLPS ρ for $\mathcal{S}$ satisfying $\mathcal{R}^* = \mathcal{R}(\rho)$ (resp. $\mathsf{post}^*(I) = \mathsf{post}(\rho, I)$).*

This *flatness* condition may seem to be a very restrictive property. However, we will later prove that most of the known semilinear classes of counter automata are in fact flat. The following lemma follows from Lemma 4.1 in [LS04], and it will be crucial to prove flatness for several classes of counter automata. Observe that this lemma is not a (direct) consequence of Parikh's Theorem, since we require the SLPS ρ to be a subset of the considered regular language L. Recall that, assuming a linear order $T = \{t_1, \ldots, t_m\}$ on T, the *Parikh map* Ψ is the total mapping from T^* to $\mathbb{N}^m$ defined by $\Psi(\pi) = (|\pi|_{t_1}, \ldots, |\pi|_{t_m})$.

Lemma 3.3. *Given a counter automaton $\mathcal{S}$, for any regular language $L \subseteq \Pi$, there exists an SLPS $\rho \subseteq L$ such that $\Psi(\rho) = \Psi(L)$.*

Accelerated symbolic model-checking consists in the usual iterative fixpoint computation, accelerated with the computation of (the effect of) some loops. In order to cope with the many variants, we analyze termination for generic versions of these accelerated reachability computations. Thus, the semi-algorithms presented below cannot be directly implemented. Effectivity issues will be discussed in Remark 3.5.

Semi-Algorithm `Accel-`$\mathcal{R}^*(\mathcal{S})$

Input:
A counter automaton $\mathcal{S}$.
Output:
The global reachability relation $\mathcal{R}^*_{\mathcal{S}}$.
 	
`let` $R \leftarrow Id_{\mathcal{C}_{\mathcal{S}}}$
`repeat forever`
  `select` one of the following tasks:
    • `if` $\mathcal{R}(T) \cdot R \subseteq R$ `return` R
    • `select` $\pi \in T^*$ and $R', R'' \subseteq R$
      `let` $R \leftarrow R \cup (R' \cdot \mathcal{R}(\pi^*) \cdot R'')$
    • `select` $t \in T$ and $R', R'' \subseteq R$
      `let` $R \leftarrow R \cup (R' \cdot \mathcal{R}(t) \cdot R'')$

Semi-Algorithm `Accel-post`$^*(\mathcal{S}, I)$

Input:
An initialized counter automaton $(\mathcal{S}, I)$.
Output:
The reachability set $\mathsf{post}^*_{\mathcal{S}}(I)$.
 	
`let` $X \leftarrow I$
`repeat forever`
  `select` one of the following tasks:
    • `if` $\mathsf{post}(T, X) \subseteq X$ `return` X
    • `select` $\pi \in T^*$ and $X' \subseteq X$
      `let` $X \leftarrow X \cup \mathsf{post}(\pi^*, X')$
    • `select` $t \in T$ and $X' \subseteq X$
      `let` $X \leftarrow X \cup \mathsf{post}(t, X')$

Theorem 3.4. *Given any counter automaton $\mathcal{S}$ and any subset $I \subseteq \mathcal{C}_{\mathcal{S}}$, we have:*

$i)$ *for every terminating execution of* `Accel-`$\mathcal{R}^*(\mathcal{S})$ *(resp.* `Accel-post`$^*(\mathcal{S}, I)$*), the returned value* `ret` *satisfies:* `ret` $= \mathcal{R}^*_{\mathcal{S}}$ *(resp.* `ret` $= \mathsf{post}^*_{\mathcal{S}}(I)$*).*
$ii)$ *there exists a terminating execution of* `Accel-`$\mathcal{R}^*(\mathcal{S})$ *(resp.* `Accel-post`$^*(\mathcal{S}, I)$*) iff $\mathcal{S}$ is globally flat (resp. $(\mathcal{S}, I)$ is flat).*

Proof. Assume that the semi-algorithm `Accel-`$\mathcal{R}^*(\mathcal{S})$ terminates. From an immediate induction over the number of times a new task is done, we deduce that there exists an SLPS ρ such that $\mathsf{post}^*(I) = \mathsf{post}(\rho, I)$. Hence $(\mathcal{S}, I)$ is flat and moreover the returned value is $\mathsf{post}^*(I)$. For the converse, assume that $(\mathcal{S}, I)$ is flat. There exists an SLPS ρ such that $\mathsf{post}^*(I) = \mathsf{post}(\rho, I)$. From this SLPS, we deduce an execution of the semi-algorithm `Accel-`$\mathcal{R}^*(\mathcal{S})$ that terminates. The proof is similar for the semi-algorithm `Accel-`$\mathcal{R}^*$. □

Remark 3.5. In order to implement these two semi-algorithms, a symbolic representation for sets of (pairs of) configurations is required. Semilinear sets are usually used since (1) they are expressive enough to express most practical flow guards, and (2) they enjoy nice decidability and closure properties. Moreover, effective acceleration results [FL02, CJ98, Boi03] can be used in order to perform the second task of the algorithm (for some classes of semilinear flow guards).

Remark 3.6. Model-checkers FAST, LASH and TREX implement "deterministic refinements" of the semi-algorithms $\mathtt{Accel}\text{-post}^*$ and $\mathtt{Accel}\text{-}\mathcal{R}^*$. FAST takes as input an initialized counter automaton in the form of a *finite-linear system*, where flow guards are given by partial integral affine transformations with semilinear definition domains. The heuristics implemented in FAST ensure termination for all flat finite-linear system [FL02].

4 Flat Counter Machines

In the remaining of this paper, we focus on a restricted class of counter automata, called counter machines, where flow guards are restricted semilinear relations given by guarded commands. Counter machines form a fairly large class of counter automata, as it contains for instance Petri nets and Minsky machines. We will show, in this section and in the next section, that many known semilinear subclasses of counter machines are flat.

First, we introduce some new notations that will be used subsequently. Recall that a *minimal element* of a subset $X \subseteq \mathbb{Q}^n$ is any $\mathsf{m} \in X$ such that for every $\mathsf{x} \in X$, if $\mathsf{x} \leq \mathsf{m}$ then $\mathsf{x} = \mathsf{m}$. We denote by $\mathrm{Min}(X)$ the *set of minimal elements* of X. It is well known that any subset of $\mathbb{N}^n$ has finitely many minimal elements [Dic13].

For every $i \in [1 \,..\, n]$, we denote $\mathbf{e}_i$ the i^{th} *basis vector* of $\mathbb{N}^n$ defined by: $\mathbf{e}_i[j] = 1$ if $j = i$ and $\mathbf{e}_i[j] = 0$ otherwise. The set $\{=, \geq\}^n$ will be considered as an alphabet, and every symbol $\# \in \{=, \geq\}^n$ will also denote the partial order on $\mathbb{Q}^n$ defined by: $\mathsf{x} \# \mathsf{y}$ if $\mathsf{x}[i] \, \#[i] \, \mathsf{y}[i]$ for all $i \in [1 \,..\, n]$.

4.1 Counter Machines

Flow guards of counter machines belong to a basic subclass of semilinear relations, called guarded commands, which we now present. An *n-dim guarded command* is any relation over $\mathbb{N}^n$ that may be written as $\{(\mathsf{x}, \mathsf{x}') \in \mathbb{N}^{2n} \;/\; \mathsf{x} \# \mu \text{ and } \mathsf{x}' = \mathsf{x} + \delta\}$ for some $\# \in \{=, \geq\}^n$, $\mu \in \mathbb{N}^n$, and $\delta \in \mathbb{Z}^n$ such that $\mu + \delta \geq \mathsf{0}$.

Remark 4.1. The class of n-dim guarded commands is the closure under composition of three kinds of basic relations:

- *increment* of a counter $i \in [1 \,..\, n]$: $\{(\mathsf{x}, \mathsf{x}') \in \mathbb{N}^{2n} \;/\; \mathsf{x}' = \mathsf{x} + \mathbf{e}_i\}$
- *decrement* of a counter $i \in [1 \,..\, n]$: $\{(\mathsf{x}, \mathsf{x}') \in \mathbb{N}^{2n} \;/\; \mathsf{x}' = \mathsf{x} - \mathbf{e}_i\}$
- 0-*test* of a counter $i \in [1 \,..\, n]$: $\{(\mathsf{x}, \mathsf{x}') \in \mathbb{N}^{2n} \;/\; \mathsf{x}[i] = 0 \text{ and } \mathsf{x}' = \mathsf{x}\}$

Definition 4.2. *An* n-dim counter machine *(counter machine for short) is an* 8*-tuple* $\mathcal{S} = (Q, T, \alpha, \beta, (G_t)_{t\in T}, \#, \mu, \delta)$*, where* $(Q, T, \alpha, \beta, (G_t)_{t\in T})$ *is a counter automaton, and where* $\# : T \to \{=, \geq\}^n$*,* $\mu : T \to \mathbb{N}^n$ *and* $\delta : T \to \mathbb{Z}^n$ *are three transition labelings satisfying:* $\mu(t)+\delta(t) \geq \mathsf{0}$ *and* $G_t = \{(\mathsf{x}, \mathsf{x}') \;/\; \mathsf{x}\,\#(t)\,\mu(t)$ *and* $\mathsf{x}' = \mathsf{x}+\delta(t)\}$ *for every* $t \in T$*.*

Transition labelings $\#$, μ and δ will be called *condition labeling*, *min labeling* and *displacement labeling* respectively. We extend the displacement labeling δ to words in the obvious way: $\delta(\varepsilon) = 0$ and $\delta(\pi \cdot t) = \delta(\pi) + \delta(t)$.

When $\#(t) \in \{\geq\}^n$ for every transition $t \in T$, we say that the counter machine $\mathcal{S}$ is *test-free*. The class of test-free counter machines is equivalent to the class of *vector addition systems with states* [HP79].

Obviously, any counter machine may be viewed as a counter automaton. In the following, we will identify a counter machine with its corresponding counter automaton. Observe that for any configurations (q, x) and (q', x') of a counter machine $\mathcal{S}$, and for any word $\pi \in T^*$, we have: $(q, \mathsf{x}) \xrightarrow{\pi} (q', \mathsf{x}')$ implies $\mathsf{x}' = \mathsf{x} + \delta(\pi)$.

The following *acceleration theorem* for counter machines, which was actually proved for larger classes of counter automata, shows that the reachability subrelation "along" any SLPS is effectively semilinear. As a direct consequence of this theorem (see for instance [LS04]), we obtain that flatness (resp. global flatness) implies effective semilinearity of the reachability set (resp. of the global reachability relation).

Theorem 4.3 ([CJ98, FL02, Boi03]). *For any SLPS* ρ *in a counter machine* $\mathcal{S}$*, the reachability subrelation* $\mathcal{R}_{\mathcal{S}}(\rho)$ *is effectively semilinear.*

Corollary 4.4. *The global reachability relation* $\mathcal{R}^*_{\mathcal{S}}$ *(resp. reachability set* $\mathsf{post}^*_{\mathcal{S}}(I)$*) of any globally flat counter machine* $\mathcal{S}$ *(resp. flat initialized counter machine* $(\mathcal{S}, I)$*) is effectively semilinear.*

Our example counter automaton $\mathcal{E}$, which actually is a counter machine, shows that the converse of this corollary does not hold (see also Remark 4.11).

Example 4.5. Recall that the counter automaton $\mathcal{E}$ introduced in Example 2.4 has a semilinear global reachability relation. In particular the reachability set $\mathsf{post}^*_{\mathcal{E}}(I)$ is semilinear for any semilinear set $I \subseteq \mathcal{C}_{\mathcal{E}}$. However, $(\mathcal{E}, (q_1, (0, 0)))$ is not flat. Intuitively, any loop $\theta \in T^*$ is either in l_1^*, l_2^*, $l_1^*\, t_1\, T^*\, t_2\, l_1^*$, or in $l_2^*\, t_2\, T^*\, t_1\, l_2^*$. In each case, we can verify that $\mathsf{post}_{\mathcal{E}}(\theta^*, I)$ is finite for any finite $I \subseteq \mathcal{C}_{\mathcal{E}}$. An induction over the length of an SLPS ρ, proves that $\mathsf{post}_{\mathcal{E}}(\rho, I)$ is finite for any finite $I \subseteq \mathcal{C}_{\mathcal{E}}$ and for any SLPS ρ. As the reachability set $\mathsf{post}^*_{\mathcal{E}}(\{(q_1, (0, 0))\}) = \{(q_1, (x, y)) \;/\; x + y \in 2\,\mathbb{N}\} \cup \{(q_2, (x, y)) \;/\; x + y - 1 \in 2\,\mathbb{N}\}$ is infinite we deduce that $(\mathcal{E}, (q_1, (0, 0)))$ is not flat.

Remark 4.6. Unfortunately, flatness is undecidable for counter machines. Indeed, the boundedness problem (is $\mathsf{post}^*_{\mathcal{S}}(\{(q, \mathsf{x}_0)\})$ finite?), which is known to be undecidable for 2-dim counter machines, is reducible to the flatness problem as follows: (1) if $(\mathcal{S}, I)$ is flat, then we can compute a semilinear description $\mathsf{post}^*_{\mathcal{S}}(I)$ and decide whether $\mathsf{post}^*_{\mathcal{S}}(I)$ is finite ; (2) if $(\mathcal{S}, I)$ is not flat, then $\mathsf{post}^*_{\mathcal{S}}(\{(q, \mathsf{x}_0)\})$ is necessarily infinite.

4.2 Reversal-bounded Counter Machines

We focus in this subsection on reversal-bounded counter machines. Intuitively, an initialized counter machine $(\mathcal{S}, I)$ will be called reversal-bounded when there exists $r \in \mathbb{N}$ such that every counter in every run of $\mathcal{S}$ from I makes at most r reversals (alternations between nondecreasing and nonincreasing modes) [Iba78]. The definition will be made precise with the use of letter morphisms.

Consider a finite set T of transitions and a displacement labeling $\delta : T \to \mathbb{Z}^n$. For every $i \in [1 \mathinner{..} n]$, we define the morphism $\varphi_i^\delta : T^* \to \{+, -\}^*$ by: $\varphi_i^\delta(t) = +$ if $\delta(t)[i] > 0$, $\varphi_i^\delta(t) = -$ if $\delta(t)[i] < 0$, and $\varphi_i^\delta(t) = \varepsilon$ if $\delta(t)[i] = 0$.

Definition 4.7. *An initialized counter machine $(\mathcal{S}, I)$, with transition set T and displacement labeling δ, is called* reversal-bounded *if there exists $r \in \mathbb{N}$ such that $\varphi_i^\delta(\pi) \in (\{+\}^* \cup \{-\}^*)^r$ for every $i \in [1 \mathinner{..} n]$ and every trace π of $\mathcal{S}$ from I. A counter machine $\mathcal{S}$ is called* globally reversal-bounded *if $(\mathcal{S}, \mathcal{C}_\mathcal{S})$ is reversal-bounded.*

Recall that the global reachability relation (resp. reachability set) of any reversal-bounded counter machine (resp. initialized counter machine) is effectively semilinear [Iba78]. We show that these two classes are flat. Note that these results do not follow from the effective semilinearity proof given in [Iba78] which uses Parikh's Theorem and manipulations on semilinear sets.

Proposition 4.8. *Every reversal-bounded initialized counter machine is flat. Every globally reversal-bounded counter machine is globally flat.*

Proof. In appendix, section A. □

4.3 Lossy/Inserting Counter Machines

Let us now focus on lossy/inserting counter machines. An n-dim counter machine will be called lossy (resp. inserting) when for every location q and for every counter $i \in [1 \mathinner{..} n]$, there is a loop[5] on q whose flow guard is the decrement (resp. increment) of counter i. Formally:

Definition 4.9. *A counter machine $\mathcal{S}$, with location set Q and transition set T, is called* lossy *(resp.* inserting*) if for every $q \in Q$ and for every $i \in [1 \mathinner{..} n]$, there exists a loop π on q such that $G_\pi = \{(\mathsf{x}, \mathsf{x}') \in \mathbb{N}^{2n} \ / \ \mathsf{x}' = \mathsf{x} - \mathbf{e}_i\}$ (resp. $G_\pi = \{(\mathsf{x}, \mathsf{x}') \in \mathbb{N}^{2n} \ / \ \mathsf{x}' = \mathsf{x} + \mathbf{e}_i\}$).*

Observe that the inverse of any lossy (resp. inserting) counter machine is an inserting (resp. lossy) counter machine. The reachability set of any initialized lossy (resp. inserting) counter machine is obviously semilinear since it is downward (resp. upward) closed (w.r.t. the usual partial order on configurations of counter automata). Moreover, it is effectively semilinear for any initialized lossy test-free counter machine and for any initialized inserting counter machine [BM99]. We show that these two classes are flat.

[5] We use an explicit representation of losses and insertions. Our flatness results given in Proposition 4.10 also hold when losses and insertions are "hardcoded" in the semantics.

Proposition 4.10. *Every initialized inserting counter machine is flat. Every initialized lossy test-free counter machine is flat.*

Proof. Consider an initialized inserting counter machine $(\mathcal{S}, I)$. It is readily seen that $\mathsf{post}^*(I) = \mathrm{Min}(\mathsf{post}^*(I)) + \mathbb{N}^n$. As $\mathrm{Min}(\mathsf{post}^*(I))$ is finite, there exists a finite set of paths $\rho_m \subseteq \Pi$ such that $\mathsf{post}(\rho_m, I) = \mathrm{Min}(\mathsf{post}^*(I))$. Moreover, for every $q \in Q$ and for every $i \in [1 \mathinner{..} n]$, there exists a loop $\pi_{q,i}$ on q such that $G_\pi = \{(\mathsf{x}, \mathsf{x}') \in \mathbb{N}^{2n} \ / \ \mathsf{x}' = \mathsf{x} + \mathbf{e}_i\}$. Consider the SLPS $\rho = \bigcup_{q \in Q} (\pi_{q,1})^* \dots (\pi_{q,n})^*$. It is readily seen that $\mathcal{R}(\rho) = \{((q, \mathsf{x}), (q, \mathsf{x}')) \in \mathcal{C}_\mathcal{S} \ / \ q \in Q \text{ and } \mathsf{x}' \geq \mathsf{x}\}$. Therefore, $\mathsf{post}^*(I) = \mathsf{post}(\rho_m\, \rho, I)$. We only sketch the proof of flatness for lossy test-free counter machines. Consider an initialized lossy test-free counter machine $(\mathcal{S}, I)$. Recall that Karp-Miller's algorithm [KM69] basically (1) computes a forward reachability tree where nodes are labeled by vectors in $(\mathbb{N} \cup \{\omega\})^n$, and (2) accelerates sequences of transitions in order to replace some components by ω. Observe that vectors in $(\mathbb{N} \cup \{\omega\})^n$ may be interpreted as downward-closed subsets of $\mathbb{N}^n$. Thus, Karp-Miller's algorithm may be seen as a "refinement" of the `Accelerated-`post^* semi-algorithm, where at each step, loops corresponding to losses are accelerated so that the current set of reachable configurations becomes downward-closed. Since Karp-Miller's algorithm always terminates, we deduce from Theorem 3.4 that $(\mathcal{S}, I)$ is flat. □

The previous proposition cannot be extended to global flatness, since there exists a 3-dim lossy test-free counter machine having a non semilinear (and hence non flat) global reachability relation [LS04]. Moreover, the test-freeness condition cannot be relaxed for lossy counter machines, since the semilinear reachability set is not in general constructible for initialized lossy counter machines [DJS99, BM99]. The following remark shows that the test-freeness condition cannot be removed even in dimension 2.

Remark 4.11. Recall that every initialized 2-dim lossy counter machine has an effectively semilinear reachability set [FS00a]. Still, there are initialized 2-dim lossy counter machines that are not flat. Consider for instance our example counter machine $(\mathcal{E}, \{(q_1, (1, 0))\})$, which is not flat according to Example 2.4, augmented with loss loops on each location: the resulting 2-dim lossy counter machine obviously remains non flat.

4.4 Test-free 2-dim Counter Machines

We briefly recall in this section known results on test-free 2-dim counter machines. The reachability set of any initialized test-free 2-dim counter machine is effectively semilinear [HP79]. Moreover, the global reachability relation is also effectively semilinear for this class [LS04]. The proof of this second result actually used flatness-based proof techniques:

Proposition 4.12 ([LS04]). *Every test-free* 2*-dim counter machine is globally flat.*

5 Flat Petri Nets

We now restrict our attention to a well-known and extensively studied subclass of counter machines: Petri nets. Usually, a Petri net is given by a directed graph whose nodes are either places or transitions. We give an equivalent definition in terms of counter machines.

Definition 5.1. *An n-dim* Petri net *(Petri net for short) is any test-free n-dim counter machine whose location set is a singleton.*

As the set Q of locations in a Petri net is a singleton, we unambiguously denote any configuration (q, x) by x.

5.1 Cyclic and Reversible Petri Nets

We focus in this subsection on two subclasses of Petri nets: *cyclic Petri nets* [AK77] and *reversible Petri nets* [Tai68]. Intuitively, an initialized Petri net will be called cyclic if its reachability set is a strongly connected component ; and a Petri net will be called reversible if every transition has an inverse.

Definition 5.2. *An initialized Petri net $(\mathcal{S}, I)$ is called* cyclic *if $I \subseteq \mathsf{post}^*(X)$ for every $X \subseteq \mathsf{post}^*(I)$. A Petri net $\mathcal{S}$ is called* globally cyclic *if $(\mathcal{S}, \mathsf{x}_0)$ is cyclic for every $\mathsf{x}_0 \in \mathcal{C}_{\mathcal{S}}$.*

Definition 5.3. *A Petri net with transition set T is called* reversible *if for every $t \in T$, there exists $t' \in T$ such that $\mathcal{R}(t') = \mathcal{R}(t)^{-1}$.*

Observe that a Petri net is globally cyclic iff its global reachability relation is symmetric iff for every transition t, there exists a path π such that $\mathcal{R}(\pi) = \mathcal{R}(t)^{-1}$. Thus, every reversible Petri net is globally cyclic. It is well-known that the global reachability relation (resp. reachability set) of any reversible Petri net (resp. cyclic initialized Petri net) is effectively semilinear [AK77, Tai68, BF97]. We show that these three classes are flat.

Proposition 5.4. *Every cyclic initialized Petri net is flat. Every globally cyclic Petri net is globally flat.*

Proof. In appendix, section B. □

Remark 5.5. Recall that global flatness implies effective semilinearity of the global reachability relation. Hence, combined with the short proof given in [Hir94] that every congruence on $\mathbb{N}^n$ is semilinear, the previous proposition gives an easy proof of effective semilinearity of $\mathcal{R}^*$ for reversible petri nets. The first proof (and only proof, to our knowledge) of this result is presented in [Tai68] and it is very difficult to read.

5.2 Regular Petri Nets

We now turn our attention to the class of regular Petri nets [VVN81]. Recall that the trace set of an initialized Petri net $(\mathcal{S}, I)$ is the set of all paths $\pi \in T^*$ such that $\mathsf{post}(\pi, I) \neq \emptyset$.

Definition 5.6. *An initialized Petri net is called* regular *if its trace set is a regular language.*

A *singly-initialized Petri net* is any initialized Petri net $(\mathcal{S}, I)$ where I is a singleton. It follows from Parikh's Theorem that the reachability set of any regular singly-initialized Petri net is effectively semilinear [VVN81]. We deduce from Lemma 3.3, which is a variant of Parikh's Theorem, that this class is actually flat.

Proposition 5.7. *Every regular singly-initialized Petri net is flat.*

Proof. Let $(\mathcal{S}, \{\mathsf{x}_0\})$ be a regular singly-initialized Petri net, and let L denote its regular trace set. According to Lemma 3.3, there exists an SLPS $\rho \subseteq L$ such that L and ρ have the same reflexive closure. Assume that $\mathsf{x}_0 \xrightarrow{\pi} \mathsf{x}$. Since $\pi \in L$, there exists $\pi' \in \rho$ such that π' is a permutation of π. Hence $\delta(\pi') = \delta(\pi)$ and as π' is a trace from x_0, we get that $\mathsf{x}_0 \xrightarrow{\pi'} \mathsf{x}_0 + \delta(\pi') = \mathsf{x}$. We conclude that $\mathsf{post}^*(\{\mathsf{x}_0\}) = \mathsf{post}(\rho, \{\mathsf{x}_0\})$. □

5.3 Persistent and Conflict-free Petri Nets

Persistent and Conflict-free Petri nets are among the first subclasses of Petri nets introduced in the literature. Intuitively, a Petri net is conflict-free if every "enabled" transition remains enabled until it is taken. For persistent Petri nets, this condition only has to hold for reachable configurations.

Definition 5.8. *An initialized Petri net $(\mathcal{S}, I)$ is called* persistent *if for any transitions t_1, t_2 with $t_1 \neq t_2$, and for any $\mathsf{x}, \mathsf{x}_1, \mathsf{x}_2 \in \mathsf{post}^*_{\mathcal{S}}(I)$ such that $\mathsf{x} \xrightarrow{t_1} \mathsf{x}_1$ and $\mathsf{x} \xrightarrow{t_2} \mathsf{x}_2$, there exists $\mathsf{x}' \in \mathsf{post}^*_{\mathcal{S}}(I)$ such that $\mathsf{x} \xrightarrow{t_1 t_2} \mathsf{x}'$.*

Definition 5.9. *A Petri net $\mathcal{S}$ is called* conflict-free *if $(\mathcal{S}, \mathcal{C}_{\mathcal{S}})$ is persistent.*

Semilinearity of the reachability set for singly-initialized persistent Petri nets was first proved in [LR78] in a non-constructive way, and a constructive proof was later presented in [May81]. It turns out that flatness, and hence effective semilinearity, can actually be deduced from the first proof. Let us first recall two lemmas from [LR78]: a weaker version of Lemma 3.1 and Lemma 4.3.

Lemma 5.10. *Given any singly-initialized persistent Petri net $(\mathcal{S}, \{\mathsf{x}_0\})$, for any two traces σ_1 and σ_2 with $\Psi(\sigma_1) \leq \Psi(\sigma_2)$, there exists a path σ' such that $\sigma_1 \sigma'$ is a trace and $\Psi(\sigma_2) = \Psi(\sigma_1) + \Psi(\sigma')$.*

Lemma 5.11. *For any singly-initialized persistent Petri net $(\mathcal{S}, \{\mathsf{x}_0\})$, there exists a finite set F of paths $\pi \in T^+$ with $\delta(\pi) \geq 0$ such that for every $\mathsf{x}_0 \xrightarrow{*} \mathsf{x} \xrightarrow{*} \mathsf{x}'$, if $\mathsf{x} \leq \mathsf{x}'$ then there exists $\pi_1, \ldots, \pi_k \in F$ such that $\mathsf{x} \xrightarrow{\pi_1 \cdots \pi_k} \mathsf{x}'$.*

Following the proof given in [LR78] that singly-initialized persistent Petri nets have semilinear reachability sets, we deduce the following theorem.

Theorem 5.12. *Every semilinearly-initialized persistent Petri net is flat.*

Proof. In appendix, section C. □

Corollary 5.13. *Every conflict-free Petri net is globally flat.*

Proof. Consider a conflict-free Petri net $\mathcal{S}$ with n counters x_1, ..., x_n. By adding to $\mathcal{S}$ n new counters x'_1, ..., x'_n that are neither tested, nor incremented, nor decremented, we obtain a new conflict-free Petri net $\mathcal{S}'$ with $2\,n$ counters. Remark that for any $L \subseteq T^*$, we have $\mathcal{R}_{\mathcal{S}}(L) = \mathsf{post}_{\mathcal{S}'}(L, I)$ where $I = \{(\mathsf{x}, \mathsf{x}') \in \mathbb{N}^{2n} \ / \ \mathsf{x} = \mathsf{x}'\}$. As $\mathcal{S}'$ is conflict-free, Theorem 5.12 proves that $(\mathcal{S}', I)$ is flat and in particular there exists an SLPS ρ such that $\mathsf{post}^*_{\mathcal{S}'}(I) = \mathsf{post}_{\mathcal{S}'}(\rho, I)$. We deduce that $\mathcal{R}^*_{\mathcal{S}} = \mathcal{R}_{\mathcal{S}}(\rho)$ and hence $\mathcal{S}$ is globally flat. $\square$

Remark 5.14. Recall that global flatness implies effective semilinearity of the global reachability relation. Hence, the we obtain that the global reachability relation is effectively semilinear for conflict-free Petri nets.

5.4 BPP-Nets

We briefly recall in this section known results on BPP-nets. An n-dim Petri net, with transition set T and min labeling μ, is called a *BPP-net* if for every $t \in T$, $\mu(t) = \mathbf{e}_i$ for some $i \in [1 \mathinner{..} n]$.

Let us recall that the global reachability relation is effectively semilinear for BPP-nets [Esp97, FO97]. The proof of this result given in [FO97] actually uses flatness-based proof techniques:

Proposition 5.15 ([FO97]). *Every BPP-net is globally flat.*

References

[ABS01] A. Annichini, A. Bouajjani, and M. Sighireanu. TReX: A tool for reachability analysis of complex systems. In *Proc. 13th Int. Conf. Computer Aided Verification (CAV'2001), Paris, France, July 2001*, volume 2102 of *Lecture Notes in Computer Science*, pages 368–372. Springer, 2001.

[AK77] T. Araki and T. Kasami. Decidable problems on the strong connectivity of Petri net reachability sets. *Theoretical Computer Science*, 4(1):99–119, 1977.

[BF97] Z. Bouziane and A. Finkel. Cyclic petri net reachability sets are semi-linear effectively constructible. In *Proc. 2nd Int. Workshop on Verification of Infinite State Systems (INFINITY'97), Bologna, Italy, July 1997*, volume 9 of *Electronic Notes in Theor. Comp. Sci.* Elsevier, 1997.

[BFLP03] S. Bardin, A. Finkel, J. Leroux, and L. Petrucci. FAST: Fast Acceleration of Symbolic Transition systems. In *Proc. 15th Int. Conf. Computer Aided Verification (CAV'2003), Boulder, CO, USA, July 2003*, volume 2725 of *Lecture Notes in Computer Science*, pages 118–121. Springer, 2003.

[BGWW97] B. Boigelot, P. Godefroid, B. Willems, and P. Wolper. The power of QDDs. In *Proc. Static Analysis 4th Int. Symp. (SAS'97), Paris, France, Sep. 1997*, volume 1302 of *Lecture Notes in Computer Science*, pages 172–186. Springer, 1997.

[BH99] A. Bouajjani and P. Habermehl. Symbolic reachability analysis of FIFO-channel systems with nonregular sets of configurations. *Theoretical Computer Science*, 221(1–2):211–250, 1999.

[BM99] A. Bouajjani and R. Mayr. Model checking lossy vector addition systems. In *Proc. 16th Ann. Symp. Theoretical Aspects of Computer Science (STACS'99), Trier, Germany, Mar. 1999*, volume 1563 of *Lecture Notes in Computer Science*, pages 323–333. Springer, 1999.

[Boi03] B. Boigelot. On iterating linear transformations over recognizable sets of integers. *Theoretical Computer Science*, 309(2):413–468, 2003.

[BW94] B. Boigelot and P. Wolper. Symbolic verification with periodic sets. In *Proc. 6th Int. Conf. Computer Aided Verification (CAV'94), Stanford, CA, USA, June 1994*, volume 818 of *Lecture Notes in Computer Science*, pages 55–67. Springer, 1994.

[CJ98] H. Comon and Y. Jurski. Multiple counters automata, safety analysis and Presburger arithmetic. In *Proc. 10th Int. Conf. Computer Aided Verification (CAV'98), Vancouver, BC, Canada, June-July 1998*, volume 1427 of *Lecture Notes in Computer Science*, pages 268–279. Springer, 1998.

[Dic13] L. E. Dickson. Finiteness of the odd perfect and primitive abundant numbers with r distinct prime factors. *Amer. Journal Math.*, 35:413–422, 1913.

[DJS99] C. Dufourd, P. Jančar, and Ph. Schnoebelen. Boundedness of Reset P/T nets. In *Proc. 26th Int. Coll. Automata, Languages, and Programming (ICALP'99), Prague, Czech Republic, July 1999*, volume 1644 of *Lecture Notes in Computer Science*, pages 301–310. Springer, 1999.

[ES69] S. Eilenberg and M.P. Schützenberger. Rational sets in commutative monoids. *Journal of Algebra*, 13:173–191, 1969.

[Esp97] J. Esparza. Petri nets, commutative context-free grammars, and basic parallel processes. *Fundamenta Informaticae*, 31(1):13–25, 1997.

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

[FL02] A. Finkel and J. Leroux. How to compose Presburger-accelerations: Applications to broadcast protocols. In *Proc. 22nd Conf. Found. of Software Technology and Theor. Comp. Sci. (FST&TCS'2002), Kanpur, India, Dec. 2002*, volume 2556 of *Lecture Notes in Computer Science*, pages 145–156. Springer, 2002.

[FO97] L. Fribourg and H. Olsén. A decompositional approach for computing least fixed-points of Datalog programs with Z-counters. *Constraints*, 2(3/4):305–335, 1997.

[FS00a] A. Finkel and G. Sutre. An algorithm constructing the semilinear $post^*$ for 2-dim reset/transfer vass. In *Proc. 25th Int. Symp. Math. Found. Comp. Sci. (MFCS'2000), Bratislava, Slovakia, Aug. 2000*, volume 1893 of *Lecture Notes in Computer Science*, pages 353–362. Springer, 2000.

[FS00b] A. Finkel and G. Sutre. Decidability of reachability problems for classes of two counters automata. In *Proc. 17th Ann. Symp. Theoretical Aspects of Computer Science (STACS'2000), Lille, France, Feb. 2000*, volume 1770 of *Lecture Notes in Computer Science*, pages 346–357. Springer, 2000.

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

[Hir94] Y. Hirshfeld. Congruences in commutative semigroups. Research report ECS-LFCS-94-291, Laboratory for Foundations of Computer Science, University of Edinburgh, UK, 1994.

[HP79] J. E. Hopcroft and J.-J. Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8(2):135–159, 1979.

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

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

[Kos82] S. R. Kosaraju. Decidability of reachability in vector addition systems. In *Proc. 14th ACM Symp. Theory of Computing (STOC'82), San Francisco, CA, May 1982*, pages 267–281, 1982.

[Las] LASH homepage. `http://www.montefiore.ulg.ac.be/~boigelot/research/lash/`.

[Lat04] L. Latour. From automata to formulas: Convex integer polyhedra. In *Proc. 19th Annual IEEE Symposium on Logic in Computer Science (LICS'04), Turku, Finland July 2004*, pages 120–129. IEEE Comp. Soc. Press, 2004.

[LR78] L.H. Landweber and E.L. Robertson. Properties of conflict-free and persistent petri nets. *Journal of the ACM*, 25(3):352–364, 1978.

[LS04] J. Leroux and G. Sutre. On flatness for 2-dimensional vector addition systems with states. In *Proc. 15th Int. Conf. Concurrency Theory (CONCUR'04), London, UK, Aug.-Sep. 2004*, volume 3170 of *Lecture Notes in Computer Science*, pages 402–416. Springer, 2004.

[May81] E. W. Mayr. Persistence of vector replacement systems is decidable. *Acta Informatica*, 15:309–318, 1981.

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

[Min67] M. L. Minsky. *Computation: Finite and Infinite Machines*. Prentice Hall, London, 1 edition, 1967.

[Tai68] M.A. Taiclin. Algorithmic problems for commutative semigroups. *Soviet Math. Doklady*, 9(1):201–204, 1968.

[VVN81] R. Valk and G. Vidal-Naquet. Petri nets and regular languages. *Journal of Computer and System Sciences*, 23(3):299–325, 1981.

A Proof of Propositition 4.8

Lemma A.1. *For any alphabet T, the language T^+ may be written as a finite union of languages of the form $t_0\, U_0^* \cdots t_{k-1}\, U_{k-1}^*\, t_k$ where:*

$i)$ each $t_j \in T$ and each $U_j \subseteq T$, and
$ii)$ for every $j \in [0 \mathinner{..} k-1]$ and for every $t \in U_j$, there exists $h \leq j < l$ such that $t = t_h = t_l$.

Proof. For any $T' \subseteq T$, let $T'_{\geq 2}$ (resp. $T'_{=2}$) denote the set of all words $\sigma \in T'^*$ such that $|\sigma|_t \geq 2$ (resp. $|\sigma|_t = 2$) for all $t \in T'$. We have the following equality[6]:

$$T^+ \quad = \quad \bigcup_{T' \subseteq T} \; \bigcup_{T'' \subseteq T \setminus T'} (T'_{\geq 2} \;⧢\; T'')$$

Thus, it is sufficient to prove that for any $T' \subseteq T$, $T'_{\geq 2}$ may be written as a finite union of languages of the required form. Now, given any word $\sigma = t_0 \ldots t_k \in T'_{=2}$ where $t_i \in T'$, and given $i \in [0 \mathinner{..} k-1]$, let U_i^σ be the set of transitions $t \in T'$ such that $|t_0 \ldots t_i|_t = 1$. Observe that we have:

$$T'_{\geq 2} = \bigcup_{\sigma = t_0 \cdots t_k \in T'_{=2}} t_0 (U_0^\sigma)^* \cdots t_{k-1} (U_{k-1}^\sigma)^* t_k$$

It is readily seen that this decomposition satisfies $i)$ and $ii)$, which concludes the proof. □

Proposition 4.8. *Every reversal-bounded initialized counter machine is flat. Every globally reversal-bounded counter machine is globally flat.*

Proof. Consider an initialized n-dim counter machine $(\mathcal{S}, I)$, with transition set T and displacement labeling δ. For every *displacement mode* $\mathsf{m} \in \{+,-\}^n$, we denote by T_m the set of transitions $t \in T$ such that $\varphi_i^\delta(t) \in \{\mathsf{m}[i], \varepsilon\}$ for every $i \in [1 \mathinner{..} n]$.

Observe that $(\mathcal{S}, I)$ is reversal-bounded (resp. $\mathcal{S}$ is globally reversal-bounded) iff there exists $r \in \mathbb{N}$ such that $\mathsf{post}^*(I) = \mathsf{post}(L^r, I)$ (resp. $\mathcal{R}^* = \mathcal{R}(L^r)$) where $L = \bigcup_{\mathsf{m} \in \{+,-\}^n} T_\mathsf{m}^*$. Hence, in order to prove the proposition, it is sufficient to show that $\mathcal{R}(T_\mathsf{m}^*)$ is flat for every $\mathsf{m} \in \{+,-\}^n$.

Consider a fixed $\mathsf{m} \in \{+,-\}^n$ and let us prove that $\mathcal{R}(T_\mathsf{m}^*) = \mathcal{R}(\rho)$ for some SLPS ρ. The proof relies on the observation that, along every path, counters are evolving in same "direction". Hence, guards have to be checked only for the first and the last occurrence of each transition. Thus, the following decomposition of T_m^* will be useful. According to Lemma A.1, the language T_m^* may be written as a finite union of languages of the form $t_0\, U_0^* \cdots t_k\, U_k^*\, t_{k+1}$ such that conditions $i)$ and $ii)$ of Lemma A.1 are satisfied.

[6] Recall that the *shuffle operator* ⧢ over languages is defined by:

$$L \;⧢\; L' = \{w_0 w'_0 \cdots w_k w'_k \;/\; w_0 \cdots w_k \in L \text{ and } w'_0 \cdots w'_k \in L'\}$$

.

Now let us consider any language $L = t_0\, U_0^* \cdots t_k\, U_k^*\, t_{k+1}$ such that conditions $i)$ and $ii)$ are satisfied. We show that $\mathcal{R}(L)$ is flat, which will conclude the proof. For every $t \in T$, we denote by $\overline{t}$ the transition obtained from t by relaxing the guard: $\overline{t}$ has the same source and target as t, and its flow guard is defined by $G_{\overline{t}} = \{(\mathsf{x}, \mathsf{x}') \in \mathbb{N}^{2n} \;/\; \mathsf{x}' = \mathsf{x} + \delta(t)\}$. It is readily seen that, for any transition $t \in T_{\mathsf{m}}$ and for any words $\pi, \pi' \in T_{\mathsf{m}}^*$, we have $\mathcal{R}(t\, \pi\, t\, \pi'\, t) = \mathcal{R}(t\, \pi\, \overline{t}\, \pi'\, t)$.

We also denote by $\overline{U_j}$ the set $\{\overline{t} \;/\; t \in U_j\}$ for each $j \in [0\,..\,k]$. We obtain that $\mathcal{R}(L) = \mathcal{R}(t_0\, \overline{U_0}^* \cdots t_k\, \overline{U_k}^*\, t_{k+1})$. For every $j \in [0\,..\,k]$, there exists according to Lemma 3.3 an SLPS $\overline{\rho_j} \subseteq \overline{U_j}^*$ such that $\mathit{\Pi} \cap \overline{U_j}^*$ and $\overline{\rho_j}$ have the same reflexive closure. Therefore, we get that:

$$\begin{aligned}\mathcal{R}(\overline{U_j}^*) &= \{((q, \mathsf{x}), (q', \mathsf{x}')) \in \mathcal{C}_{\mathcal{S}} \;/\; \exists \pi \in (\mathit{\Pi}(q, q') \cap \overline{U_j}), \mathsf{x}' = \mathsf{x} + \delta(\pi)\} \\ &= \mathcal{R}(\overline{\rho_j})\end{aligned}$$

since for every $i \in [1\,..\,n]$, all the $\delta(\overline{t})[i]$, with $\overline{t} \in \overline{U_j}$, have the same sign. Consequently, we get:

$$\begin{aligned}\mathcal{R}(L) &= \mathcal{R}(t_0\, U_0^* \cdots t_k\, U_k^*\, t_{k+1}) \\ &= \mathcal{R}(t_0\, \overline{U_0}^* \cdots t_k\, \overline{U_k}^*\, t_{k+1}) \\ &= \mathcal{R}(t_0) \cdot \mathcal{R}(\overline{U_0}^*) \cdots \mathcal{R}(t_k) \cdot \mathcal{R}(\overline{U_k}^*) \cdot \mathcal{R}(t_{k+1}) \\ &= \mathcal{R}(t_0) \cdot \mathcal{R}(\overline{\rho_0}) \cdots \mathcal{R}(t_k) \cdot \mathcal{R}(\overline{\rho_k}) \cdot \mathcal{R}(t_{k+1}) \\ &= \mathcal{R}(t_0\, \overline{\rho_0} \cdots t_k\, \overline{\rho_k}\, t_{k+1}) \\ &= \mathcal{R}(t_0\, \rho_0 \cdots t_k\, \rho_k\, t_{k+1})\end{aligned}$$

where, for every $j \in [0\,..\,k]$, we denote by ρ_j the SLPS obtained from $\overline{\rho_j}$ by replacing each $\overline{t}$ by t (note that $\rho_j \subseteq U_j^*$, which justifies the last equality in the above equations). □

B Proof of Proposition 5.4

Proposition 5.4. *Every cyclic initialized Petri net is flat. Every globally cyclic Petri net is globally flat.*

Proof. Let $(\mathcal{S}, I)$ be a cyclic initialized n-dim Petri net, and let $\mathsf{x}_0 \in I$. It is readily seen that $\mathsf{post}^*(I) = \mathsf{post}^*(\{\mathsf{x}_0\})$. From Theorem 3.5 in [BF97], we get that:

$$\mathsf{post}^*(I) \;\;=\;\; \mathrm{Min}(\mathsf{post}^*(\{\mathsf{x}_0\})) \;+\; (\mathrm{Min}((\mathsf{post}^*(\mathsf{x}_0) - \mathsf{x}_0) \,\cap\, \mathbb{N}^n))^*$$

For every $\mathsf{m} \in \mathrm{Min}(\mathsf{post}^*(I))$ and $\mathsf{p} \in \mathrm{Min}((\mathsf{post}^*(\mathsf{x}_0) - \mathsf{x}_0) \cap \mathbb{N}^n)$, there exists $\pi_{\mathsf{m}} \in T^*$ such that $\mathsf{x}_0 \xrightarrow{\pi_{\mathsf{m}}} \mathsf{m}$ and there exists $\pi'_{\mathsf{p}} \in T^*$ such that $\mathsf{x}_0 \xrightarrow{\pi'_{\mathsf{p}}} \mathsf{x}_0 + \mathsf{p}$. The SLPS witnessing flatness of $(\mathcal{S}, I)$ is $\rho' \cdot \rho$, where ρ is the union of the π_{m}, and ρ' is the concatenation (in any order) of the $(\pi'_{\mathsf{p}})^*$.

Let $\mathcal{S}$ be a globally cyclic n-dim Petri net. As $\mathcal{R}^*$ is symmetric, we obtain that $\mathcal{R}^*$ is a congruence on $\mathbb{N}^n$ and hence it is semilinear [ES69]. Therefore, it is sufficient to prove that every linear set contained in $\mathcal{R}^*$ is also contained in $\mathcal{R}(\rho)$ for some SLPS ρ. Consider a basis $(\mathsf{x}, \mathsf{x}') \in \mathbb{N}^{2n}$ and a finite set of periods $P = \{(\mathsf{p}_1, \mathsf{p}'_1), \ldots, (\mathsf{p}_k, \mathsf{p}'_k)\} \subseteq \mathbb{N}^{2n}$ and assume that $(\mathsf{x} + P^*) \subseteq \mathcal{R}^*$. There exists $\pi_0, \pi_1, \ldots, \pi_k$ in T^* such that $\mathsf{x} \xrightarrow{\pi_0} \mathsf{x}'$, and $\mathsf{x} + \mathsf{p}_i \xrightarrow{\pi_i} \mathsf{x}' + \mathsf{p}'_i$ for every $i \in [1\,..\,k]$. As $\mathcal{R}^*$ is symmetric, there exits $\overline{\pi_0} \in T^*$ such that $\mathsf{x}' \xrightarrow{\overline{\pi_0}} \mathsf{x}$. Consider the LPS $\rho = (\pi_1 \overline{\pi_0})^* \ldots (\pi_k \overline{\pi_0})^* \cdot \pi_0$. It is readily seen that $\mathcal{R}(\rho) \supseteq (\mathsf{x}, \mathsf{x}') + P^*$. □

C Proof of Theorem 5.12

Theorem 5.12. *Every semilinearly-initialized persistent Petri net is flat.*

Proof. Let us first prove that any singly-initialized persistent Petri net $(\mathcal{S}, I)$ with $I = \{\mathsf{x}_0\}$ if flat. Consider a finite set $F = \{\pi_1, \ldots, \pi_m\}$ of paths in T^+ satisfying Lemma 5.11 and let us first prove that $\rho_+ = (\pi_1^* \ldots \pi_m^*)^n$ is a SLPS such that for any $\mathsf{x}_0 \xrightarrow{*} \mathsf{x} \xrightarrow{*} \mathsf{x}'$, if $\mathsf{x} \leq \mathsf{x}'$ then $(\mathsf{x}, \mathsf{x}') \in \mathcal{R}_{\mathcal{S}}(\rho_+)$. From Lemma 5.11, there exists a sequence π_{i_1}, ..., π_{i_k} in F such that $\mathsf{x} \xrightarrow{\pi_{i_1} \ldots \pi_{i_k}} \mathsf{x}'$. As $\delta(\pi) \geq 0$ for any $\pi \in F$, the word $\pi_{i_1} \ldots \pi_{i_k}$ can be reordered into a word w such that $\mathsf{x} \xrightarrow{w} \mathsf{x}'$ and $w = \pi_{j_1}^{\alpha_1} \ldots \pi_{j_m}^{\alpha_m}$ where $\{j_1, \ldots, j_m\} = \{1, \ldots, n\}$ and $\alpha_i \geq 0$. From $\pi_{j_m}^{\alpha_m} \in \pi_1^* \ldots \pi_m^*$, we deduce that $w \in \rho_+$ and we get $(\mathsf{x}, \mathsf{x}') \in \mathcal{R}_{\mathcal{S}}(\rho_+)$. Now, let M denote the set $\mathrm{Min}(\{(\mathsf{x}, \Psi(\sigma)) \;/\; \mathsf{x}_0 \xrightarrow{\sigma} \mathsf{x}\})$. Recall that M is ***finite*** and in particular, there exists a finite subset Σ_0 of the trace set of $(\mathcal{S}, I)$ such that $M = \{(\mathsf{x}_0 + \delta(\sigma), \Psi(\sigma)) \;/\; \sigma \in \Sigma_0\}$. Let us prove that $\mathsf{post}_{\mathcal{S}}^*(I) = \mathsf{post}_{\mathcal{S}}(\rho, I)$ where $\rho = \Sigma_0 \cdot \rho_+$. Assume that $\mathsf{x}_0 \xrightarrow{\sigma'} \mathsf{x}'$. By definition of M, there exists $\mathsf{x}_0 \xrightarrow{\sigma} \mathsf{x}$ with $\sigma \in \Sigma_0$ such that $(\mathsf{x}, \Psi(\sigma)) \leq (\mathsf{x}', \Psi(\sigma'))$. According to Lemma 5.10, there exists a path σ'' such that $\sigma\,\sigma''$ is a trace and $\Psi(\sigma') = \Psi(\sigma) + \Psi(\sigma'')$. We get that $\mathsf{x}_0 \xrightarrow{\sigma} \mathsf{x} \xrightarrow{\sigma''} \mathsf{x}_0 + \delta(\sigma) + \delta(\sigma'') = \mathsf{x}_0 + \delta(\sigma') = \mathsf{x}'$. Since $\mathsf{x} \xrightarrow{*} \mathsf{x}'$ and $\mathsf{x} \leq \mathsf{x}'$, we obtain that $(\mathsf{x}, \mathsf{x}') \in \mathcal{R}_{\mathcal{S}}(\rho_+)$. We have proved that $\mathsf{x}' \in \mathsf{post}_{\mathcal{S}}(\rho, I)$. We conclude that any singly-initialized persistent Petri net is flat.

Now, let us prove that any linearly-initialized persistent Petri Net $(\mathcal{S}, I)$ is flat. Recall that a *linear set* I is a set of the form $I = (\mathsf{x}_0 + P^*)$ where $\{\mathsf{x}_0\} \cup P$ is a finite subset of $\mathcal{C}_{\mathcal{S}}$. By adding to the Petri Net $\mathcal{S}$ a transition t_p such that $\mu(t_\mathsf{p}) = 0$ and $\delta(t_\mathsf{p}) = \mathsf{p}$ for each period $\mathsf{p} \in P$, we obtain a singly-initialized Petri Net $(\mathcal{S}', I')$ where $I' = \{\mathsf{x}_0\}$. Remark that $(\mathcal{S}, I)$ and $(\mathcal{S}', I')$ have the same reachability set and $(\mathcal{S}', I')$ is persistent. Therefore $(\mathcal{S}', I')$ is flat and there exists a SLPS ρ' over $T' = T \cup \{t_\mathsf{p} \;/\; \mathsf{p} \in P\}$, where T is the transition set of $\mathcal{S}$, such that $\mathsf{post}_{\mathcal{S}'}^*(I') = \mathsf{post}_{\mathcal{S}'}(\rho', \{\mathsf{x}_0\})$. By removing from ρ' all letters that are not in T, we get a SLPS ρ such that $\mathsf{post}_{\mathcal{S}}^*(I) = \mathsf{post}_{\mathcal{S}}(\rho, I)$. Therefore $(\mathcal{S}, I)$ is flat. Finally, flatness for semilinearly-initialized persistent Petri nets follows from the fact that SLPS are closed under finite union. □