



HAL
open science

Flat Petri Nets (Invited Talk)

Jérôme Leroux

► **To cite this version:**

Jérôme Leroux. Flat Petri Nets (Invited Talk). Application and Theory of Petri Nets and Concurrency - 42nd International Conference, PETRI NETS 2021, Jun 2021, Virtual Event, France. 10.1007/978-3-030-76983-3_2 . hal-03436231

HAL Id: hal-03436231

<https://hal.science/hal-03436231>

Submitted on 19 Nov 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.

Flat Petri Nets (Invited Talk)

Jérôme Leroux¹[0000-0002-7214-9467]*

Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800
<https://www.labri.fr/~leroux/>

Abstract. Vector addition systems with states (VASS for short), or equivalently Petri nets are one of the most popular formal methods for the representation and the analysis of parallel processes. The central algorithmic problem is reachability: whether from a given initial configuration there exists a sequence of valid execution steps that reaches a given final configuration. This paper provides an overview of results about the reachability problem for VASS related to Presburger arithmetic, by presenting 1) a simple algorithm for deciding the reachability problem based on invariants definable in Presburger arithmetic, 2) the class of flat VASS for computing reachability sets in Presburger arithmetic, and 3) complexity results about the reachability problem for flat VASS.

Keywords: Formal Methods · Petri Nets · Flat Systems · Presburger Arithmetic.

1 Introduction

Vector addition systems with states [30], or equivalently vector addition systems [31], or Petri nets are one of the most popular formal methods for the representation and the analysis of parallel processes [24]. The central algorithmic problem is reachability: whether from a given initial configuration there exists a sequence of valid execution steps that reaches a given final configuration. Many computational problems reduce to this reachability problem in logic, complexity, real-time systems, protocols [49, 28].

A d -dimensional vector addition system (d -VASS, or just VASS when the dimension d is not relevant) is a pair $V = (Q, T)$ where Q is a non empty finite set of elements called *states*, and T is a finite set of triples in $Q \times \mathbb{Z}^d \times Q$ called *transitions*. A *configuration* is a pair $(q, x) \in Q \times \mathbb{N}^d$ also denoted as $q(x)$ in the sequel, and an *action* is a vector in \mathbb{Z}^d . The semantics is defined by introducing for each transition t the binary relation \xrightarrow{t} over the configurations defined by $p(x) \xrightarrow{t} q(y)$ if $t = (p, y - x, q)$. We also associate to a word $\sigma = t_1 \dots t_k$ of

* The author is supported by the grant ANR-17-CE40-0028 of the French National Research Agency ANR (project BRAVAS)

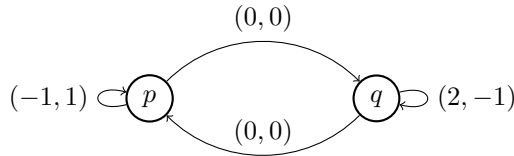
transitions t_1, \dots, t_k the binary relation $\xrightarrow{\sigma}$ over the configurations defined by $p(x) \xrightarrow{\sigma} q(y)$ if there exists a sequence c_0, \dots, c_k of configurations such that:

$$p(x) = c_0 \xrightarrow{t_1} c_1 \cdots \xrightarrow{t_k} c_k = q(y)$$

The *reachability set* from a set C_{in} of configurations is the set $\text{Reach}_V(C_{\text{in}})$ of configurations c such that $c_{\text{in}} \xrightarrow{\sigma} c$ for some configuration $c_{\text{in}} \in C_{\text{in}}$ and some word σ of transitions.

Example 1. Let us consider the VASS V depicted below. This VASS has a loop on state p and another loop on state q . Intuitively, iterating the loop on state p transfers the content of the first counter to the second counter whereas iterating the loop on state q transfers and multiplies by two the content of the second counter to the first counter. Let us denote by t_1, t_2, t_3 and t_4 the transitions $(p, (-1, 1), p)$, $(p, (0, 0), q)$, $(q, (2, -1), q)$ and $(q, 0, 0, p)$. We can prove that the reachability set from $\{p(1, 0)\}$ is equal to $\{p, q\} \times \{(n, m) \mid n + m \geq 1\}$ by observing that if $n, m \in \mathbb{N}$ satisfy $n + m \geq 1$ then:

$$p(1, 0) \xrightarrow{(t_1 t_2 t_3 t_4)^{n+m-1}} p(n+m, 0) \xrightarrow{t_1^m} p(n, m) \xrightarrow{t_2} q(n, m)$$



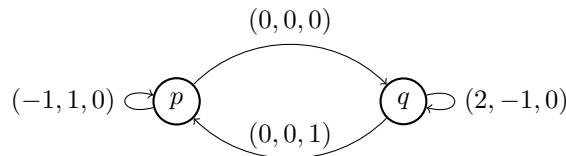
The reachability problem takes as input a VASS V and two configurations $c_{\text{in}}, c_{\text{out}}$ and it decides if there exists a word σ of transitions such that $c_{\text{in}} \xrightarrow{\sigma} c_{\text{out}}$. After an incomplete proof by Sacerdote and Tenney [48], decidability of the problem was established by Mayr [44, 45], whose proof was then simplified by Kosaraju [32]. Building on the further refinements made by Lambert in the 1990s [34], in 2015, a first complexity upperbound of the reachability problem was provided [39] more than thirty years after the presentation of the algorithm introduced by Mayr [44, 45, 32, 34]. The upperbound given in that paper is cubic Ackermannian. This complexity is obtained by analyzing the computation complexity of the Mayr algorithm. By refining this algorithm and by introducing a new ranking function proving the termination of this refinement, an Ackermannian complexity upperbound was obtained in [40]. This paper also showed that the reachability problem in fixed dimension is primitive recursive by bounding the length of executions thanks to the Grzegorzcyk hierarchy. Based on this bound, in [43], the reachability problem for general VASS is shown to be inter-reducible in log-space to the reachability problem for structurally bounded VASS when numbers are encoded in unary or in binary. Let us recall that a VASS is said to be *structurally bounded* if the reachability set is finite from any initial configuration, and this property is decidable in polynomial time (even when numbers are encoded in binary) thanks to the Kosaraju-Sullivan algorithm [33].

The reachability problem for structurally bounded VASS can be decided by a deterministic brute-force exploration in an obvious way. The computational complexity of such an algorithm is known to be Ackermannian [47]. Moreover, due to the family of VASS introduced in [46] this bound is tight. It follows that the reachability problem for general VASS can be solved with a simple deterministic brute-force algorithm, and last but not least, the reachability problem for structurally bounded VASS is a central problem.

In this paper, we present results about the reachability problem for VASS related to *Presburger arithmetic* $\text{fo}(\mathbb{N}, +)$. In this context, a set C of configurations in $Q \times \mathbb{N}^d$ is said to be *Presburger* if there exists a sequence $(\phi_q)_{q \in Q}$ of formulas ϕ_q in Presburger arithmetic denoting sets $X_q \subseteq \mathbb{N}^d$ such that $C = \bigcup_{q \in Q} \{q\} \times X_q$. In Sec. 2 we present a simple algorithm for deciding the reachability problem for VASS based on Presburger inductive invariant that shows that the Presburger sets of configurations are central for deciding the reachability problem for VASS even if, as shown in Ex. 2, there exists VASS with non Presburger reachability sets. In Sec. 3 we show that the reachability set of a VASS is Presburger, if and only if, it is flattable, i.e. the VASS can be unfolded into a VASS without nested cycles called *flat VASS*. In Sec. 4 we present complexity results about the reachability problem for flat VASS.

Example 2. In 1979, Hopcroft and Pansiot [30] introduced the VASS depicted below. This VASS exhibit a non Presburger reachability set from the initial configuration $p(1, 0, 0)$. Intuitively, on the first and the second counters, the behaviour of that VASS is the same as the one introduced in Ex. 1. The third counter is incremented each time we come back to state p from q . In [30] the reachability set from the initial configuration $p(1, 0, 0)$ is proved equal to the following set:

$$\{p(x_1, x_2, x_3) \mid x_1 + x_2 \leq 2^{x_3}\} \cup \{q(x_1, x_2, x_3) \mid x_1 + 2x_2 \leq 2^{x_3+1}\}$$



2 Presburger Inductive Invariants

We present in this section a simple algorithm for deciding the reachability problem based on Presburger inductive invariants [35, 36, 37] that may have an optimal complexity (this is an open problem). A set C of configurations is called an *inductive invariant* for a VASS V if for every configurations c, c' and every transition t such that $c \xrightarrow{t} c'$, then $c \in C$ implies $c' \in C$.

Theorem 1 ([35]). *For every VASS V , for every Presburger sets of configurations $C_{\text{in}}, C_{\text{out}}$, either $c_{\text{in}} \xrightarrow{\sigma} c_{\text{out}}$ for some configurations $c_{\text{in}} \in C_{\text{in}}$ and $c_{\text{out}} \in C_{\text{out}}$ and some word σ of transitions, or there exists a Presburger inductive invariant C that contains C_{in} and disjoint from C_{out} .*

Since we can decide if a sequence of Presburger formulas denotes an inductive invariant with classical algorithms deciding Presburger arithmetic, the previous theorem shows that a brute-force non-deterministic exploration of the reachability set and sequences of Presburger formulas provides a simple algorithm for deciding the reachability problem. Whereas the proof in [35] was based on a refinement of Lambert’s algorithm, in [36] a direct proof based on a well quasi order over the executions is provided. This proof was then simplified a bit more in a paper [37] that received a best paper award at Alan Turing centenary conference in 2012. In those two last papers, Presburger formulas denoting inductive invariants are obtained by proving that reachability sets are “asymptotically” definable in Presburger arithmetic.

Example 3. Let us come back to Ex. 2. Notice that the reachability set from the initial configuration $p(1, 0, 0)$ is not Presburger. Let us introduce the non-decreasing sequence $(C_n)_{n \in \mathbb{N}}$ of Presburger sets defined as follows:

$$C_n = \{p(x_1, x_2, x_3) \mid \bigvee_{i=0}^n (x_1 + x_2 \leq 2^i \wedge x_3 = i) \vee x_3 > n\} \cup \\ \{q(x_1, x_2, x_3) \mid \bigvee_{i=0}^n x_1 + 2x_2 \leq 2^{i+1} \wedge x_3 = i) \vee x_3 > n\}$$

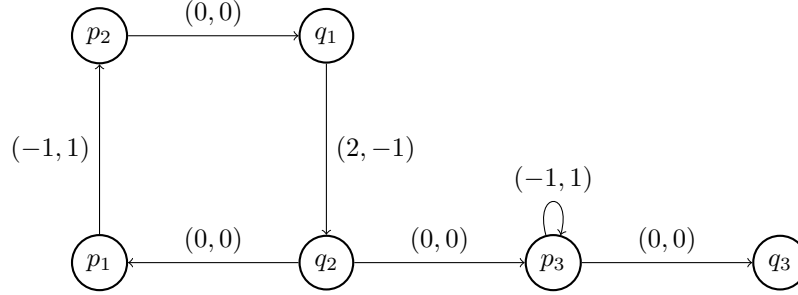
Notice that C_n is an inductive invariant that contains the initial configuration $p(1, 0, 0)$ and since $\bigcap_{n \in \mathbb{N}} C_n$ is the reachability set from $p(1, 0, 0)$, it follows that for every configuration c_{out} outside of this reachability set, there exists n such that $c_{\text{out}} \notin C_n$.

3 Flat and Flattable VASS

When the reachability set of a VASS is infinite from an initial configuration, a brute-force exploration of the reachability set fails. However, even in that case the computation of the reachability sets may still be possible by using Presburger arithmetic for symbolically representing infinite sets of configurations and by using acceleration techniques to discover infinite sets of reachable configurations. Intuitively, acceleration techniques consist in computing symbolically the effect of iterating cycles of the system. Those techniques were studied for several models: systems with FIFO channels [11, 15, 16, 10, 9], time[3, 2, 12, 13], other data structures[17], and systems manipulating counters including the VASS model [14, 27, 8, 5, 4, 6, 18, 19].

Acceleration techniques for VASS are related to the class of *flat VASS*. Formally, a VASS V is said to be *flat* if for every state q , there exists at most one simple cycle on q (intuitively no nested cycles).

Example 4. The two VASS depicted in Ex. 1 and Ex. 2 are not flat. The VASS depicted below is flat.



The reachability set of a flat VASS is clearly Presburger by compiling in Presburger arithmetic the effect of iterating simple cycles [27]. The problem of deciding if the reachability set of a general (non flat) VASS from an initial Presburger set of configurations is Presburger was studied thirty years ago independently by Dirk Hauschildt during his PhD[29] and Jean-Luc Lambert. Unfortunately, these two works were never published. Moreover, from these works, it is difficult to derive a simple algorithm for computing Presburger formulas denoting the reachability set. In [38] a simple algorithm for computing such a formula based on flat VASS is given. Intuitively when a VASS V is not flat, one can try to unfold it into a flat VASS V' such that the reachability set of V from a Presburger set C_{in} can be derived from the reachability set of V' from a Presburger set C'_{in} derived from C_{in} .

More formally, an *unfolding* of a VASS $V = (Q, T)$ is a pair (V', f) where $V' = (Q', T')$ is a VASS and $f : Q' \rightarrow Q$ is a total mapping such that $(f(p'), a, f(q'))$ is in T for every transition $(p', a, q') \in T'$. We observe that for every set of configurations C_{in} of V , we have where f is extended over the configurations by $f(q', x) = (f(q'), x)$ for every configuration $(q', x) \in Q' \times \mathbb{N}^d$:

$$f(\text{Reach}_{V'}(f^{-1}(C_{\text{in}}))) \subseteq \text{Reach}_V(C_{\text{in}})$$

When the previous inclusion is an equality, the unfolding is said to be *complete* from C_{in} . An unfolding is called a *flattening* when V' is flat [7]. A VASS V is said to be *flattable* from a set C_{in} of initial configurations if there exists a flattening of V complete from C_{in} (see [42] for various examples of flattable VASS).

Theorem 2 ([38]). *For every VASS V , for every Presburger set C_{in} of configurations, the reachability set $\text{Reach}_V(C_{\text{in}})$ is Presburger if, and only if, V is flattable from C_{in} .*

It follows that if the reachability set from an initial Presburger set of configurations is Presburger, a sequence of Presburger formulas denoting the reachability set can be computed by finding the right flattening. In [26] heuristics and algorithms for finding such a flattening are presented. Those heuristics are implemented in the tool FAST [5, 4, 6] for analyzing Minsky machines, a class of systems strictly extending VASS with undecidable reachability problem.

Remark 1. In [38], a stronger version of Thm. 2 is proved. More precisely, it is shown that for every VASS V , for every Presburger set C_{in} of configurations, and for every Presburger set $C \subseteq \text{Reach}_V(C_{\text{in}})$, there exists a flattening (V', f) of V such that $C \subseteq f(\text{Reach}_{V'}(f^{-1}(C_{\text{in}})))$. This extension is used in [25] in order to provide witnesses of well-specification for population protocols [1].

Example 5. Let $V = (Q, T)$ be the VASS introduced in Ex. 1, $V' = (Q', T')$ be the flat VASS introduced in Ex. 4, and $f : Q' \rightarrow Q$ defined by $f(p_i) = p$ and $f(q_i) = q$ for every $i \in \{1, 2, 3\}$. Observe that (V', f) is a flattening of V . Moreover, we derive from Ex. 1 that this flattening is complete from $\{p(1, 0)\}$. It follows that V is flattable from $\{p(1, 0)\}$.

Flattening are also used for deriving fine complexity results for the reachability problem for 2-VASS. Recall that the reachability sets from an initial configuration have been shown to be effectively Presburger for 2-VASS in [30]. In [41], it was proved that for every 2-VASS V there exists a flattening (V', f) of V effectively computable such that for every configuration $p(x), q(y)$, we have $p(x) \xrightarrow{*}_V q(y)$ if, and only if, there exist two states $p' \in f^{-1}(p)$ and $q' \in f^{-1}(q)$ such that $p'(x) \xrightarrow{*}_{V'} q'(y)$. Based on a similar proof, ten years later, it was proved that for every 2-VASS V , there exists a family \mathcal{F} of flattening (V', f) of V of “small sizes” such that for every configurations $p(x), q(y)$, we have $p(x) \xrightarrow{*}_V q(y)$ if, and only if, there exists a flattening (V', f) in \mathcal{F} and states $p' \in f^{-1}(p)$ and $q' \in f^{-1}(q)$ such that $p'(x) \xrightarrow{*}_{V'} q'(y)$. From this result the reachability problem for 2-VASS encoded in binary was proved to be PSPACE-complete in the same paper. Finally, thanks to the family \mathcal{F} , and the fact that the reachability problem for flat 2-VASS encoded in unary is NL-complete [23], the reachability problem for general 2-VASS encoded in unary was proved NL-complete in [23].

4 Reachability Problem for Flat VASS

In this section we present some complexity lowerbounds of the reachability problem for flat VASS. In that context, the subclass of ultraflat VASS will play a central role. Formally, an *ultraflat VASS* is a VASS $V = (Q, T)$ such that $Q = \{q_1, \dots, q_n\}$ with $n = |Q|$, and $T = \{(q_{j-1}, (0, \dots, 0), q_j) \mid 2 \leq j \leq n\} \cup \{(q_j, a_j, q_j) \mid 1 \leq j \leq n\}$ for some actions a_1, \dots, a_n . An ultraflat VASS is clearly flat since (q_j, a_j, q_j) is the unique simple cycle on q_j for every j .

The reachability problem for flat 1-VASS with numbers encoded in binary can be easily proved NP-hard by reduction of the subset sum problem. Let us recall that the subset sum problem takes as input a sequence s, s_1, \dots, s_k of natural numbers encoded in binary and it decides if there exists a finite set $I \subseteq \{1, \dots, k\}$ such that $s = \sum_{i \in I} s_i$. The following lemma shows that this lowerbound also holds for ultraflat 1-VASS in binary.

Lemma 1. *The reachability problem for ultraflat 1-VASS with numbers encoded in binary is NP-hard.*

Proof. Let us consider an instance s, s_1, \dots, s_k of the subset sum problem. We can assume that $0 < s < \sum_{j=1}^k s_j$ and $k \geq 2$ since other instances are trivially accepting or non accepting. We introduce $x = \sum_{j=1}^k (1 + s_j)$. Notice that $x \geq 4$, $s + k < x$, and $1 + s_j < x$ for every $1 \leq j \leq k$.

We introduce the ultraflat 1-VASS $V = (Q, T)$ defined by the set of states $Q = \{p_1, q_1, \dots, p_k, q_k\}$, and the set of transitions T that contains the transitions labeled by 0 that connect the states of Q to form an ultraflat VASS, and the transitions $\alpha_j = (p_j, u_j, p_j)$ and $\beta_j = (q_j, v_j, q_j)$ with $v_j = 1 - (x-1)x^{2k+1-j}$ and $u_j = s_j + v_j$ for every $1 \leq j \leq k$. Let us prove that $p_1(x^{2k+1}) \xrightarrow{*} q_k(x^{k+1} + s + k)$ if, and only if, there exists $J \subseteq \{1, \dots, k\}$ such that $s = \sum_{j \in J} s_j$.

Assume first that $p_1(x^{2k+1}) \xrightarrow{*} q_k(x^{k+1} + s + k)$. Since V is an ultraflat VASS, there exist sequences $(n_j)_{1 \leq j \leq k}$, $(m_j)_{1 \leq j \leq k}$, $(a_j)_{1 \leq j \leq k+1}$, and $(b_j)_{1 \leq j \leq k}$ of natural numbers with $a_1 = x^{2k+1}$, $a_{k+1} = x^{k+1} + s + k$, and such that for every $1 \leq j \leq k$, we have:

$$p_j(a_j) \xrightarrow{\alpha_j^{m_j}} p_j(b_j) \quad q_j(b_j) \xrightarrow{\beta_j^{n_j}} q_j(a_{j+1})$$

It follows that $a_{j+1} = a_j + m_j u_j + n_j v_j$ for every $1 \leq j \leq k$.

From $1 + s_j < x$, we derive $u_j < x - (x-1)x^{k+1} \leq (2-x)x^{k+1} \leq -x^{k+1}$ since $x \geq 3$. As $v_j \leq u_j$, we have proved that $u_j, v_j < -x^{k+1}$. Since $a_{k+1} = a_1 + \sum_{j=1}^k m_j u_j + n_j v_j \leq x^{2k+1} - (x^{k+1} + 1) \sum_{j=1}^k (m_j + n_j)$, and $a_{k+1} \geq 0$, we get $\sum_{j=1}^k m_j + n_j < x^k$. Moreover, since $u_j = 1 + s_j \pmod{x^{k+1}}$ and $v_j = 1 \pmod{x^{k+1}}$, we deduce from $a_{j+1} = a_j + m_j u_j + n_j v_j$ that $a_{j+1} = a_j + m_j(1 + s_j) + n_j \pmod{x^{k+1}}$. It follows that $a_{k+1} = a_1 + r \pmod{x^{k+1}}$ where $r = \sum_{j=1}^k (m_j(1 + s_j) + n_j)$. As $a_{k+1} = s + k \pmod{x^{k+1}}$ and $a_1 = 0 \pmod{x^{k+1}}$ we get $r = s + k \pmod{x^{k+1}}$. Since $r \leq (x^k - 1) \sum_{j=1}^k (1 + s_j) < x^{k+1}$ and $s + k < x \leq x^{k+1}$ we deduce that $r = s + k$. In particular $\sum_{j=1}^k (m_j + n_j) \leq s + k$.

Assume by contradiction that there exists $\ell \in \{1, \dots, k\}$ such that $m_\ell + n_\ell \neq 1$ and let ℓ be the minimal one. By induction we deduce that $a_\ell = x^{2k+2-\ell} + \ell - 1 + \sum_{j \in J} s_j$ where $J = \{j \in \{1, \dots, \ell-1\} \mid m_j = 1\}$. It follows that $a_\ell < x^{2k+2-\ell} + x$.

Notice that if $m_\ell + n_\ell \geq 2$ then:

$$\begin{aligned} a_{\ell+1} &= a_\ell + m_\ell u_\ell + n_\ell v_\ell \\ &\leq a_\ell + (m_\ell + n_\ell) u_\ell \\ &< x^{2k+2-\ell} + x + 2(x - (x-1)x^{2k+1-\ell}) \\ &\leq x^{2k+1-\ell}(2-x) + 3x \\ &\leq -x^{k+1} + x^2 \\ &\leq 0 \end{aligned}$$

And we get a contradiction with $a_{\ell+1} \geq 0$. Therefore $m_\ell + n_\ell \leq 1$ and since $m_\ell + n_\ell \neq 1$ we deduce that $m_\ell = n_\ell = 0$. It follows that $a_{\ell+1} = a_\ell$. In particular $a_{\ell+1} \geq x^{2k+2-\ell}$.

Now, observe that $a_{k+1} = a_{\ell+1} + \sum_{j=\ell+1}^k n_j u_j + m_j v_j \geq a_{\ell+1} + \sum_{j=\ell+1}^k (n_j + m_j) v_{\ell+1} \geq a_{\ell+1} + (s+k)v_{\ell+1}$. It follows that $a_{k+1} \geq x^{2k+2-\ell} + (s+k)(1 - (x-1)x^{2k-\ell}) = x^{2k-\ell}(x^2 - (s+k)(x-1)) + s+k$. Since $s+k \leq x-1$, we deduce that $x^2 - (s+k)(x-1) \geq x^2 - (x-1)^2 = 2x-1 > x$ since $x \geq 2$. In particular $a_{k+1} > x^{k+1} + s+k = a_{k+1}$ and we get a contradiction.

It follows that $m_j + n_j = 1$ for every $1 \leq j \leq k$. Let us introduce $J = \{j \in \{1, \dots, k\} \mid m_j = 1\}$. An immediate induction shows that $a_{k+1} = x^{k+1} + k + \sum_{j \in J} s_j$. Since $a_{k+1} = x^{k+1} + k + s$, we get $s = \sum_{j \in J} s_j$.

Conversely, observe that if there exists $J \subseteq \{1, \dots, k\}$ such that $s = \sum_{j \in J} s_j$ then $p_1(x^{2k+1}) \xrightarrow{*} q_k(x^{k+1} + s+k)$ by considering the sequence $(m_j)_{1 \leq j \leq k}$ and $(n_j)_{1 \leq j \leq k}$ satisfying $(m_j, n_j) = (1, 0)$ if $j \in J$ and $(m_j, n_j) = (0, 1)$ otherwise. \square

When the dimension is part of the input, the following lemma shows that the reachability problem for ultraflat VASS in unary is also NP-hard.

Lemma 2. *The reachability problem for ultraflat VASS with numbers encoded in unary is NP-hard.*

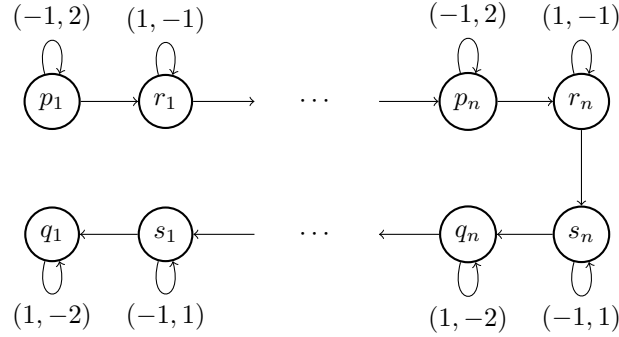
Proof. Let us consider an instance s, s_1, \dots, s_k of the subset sum problem. We can assume that $s \leq \sum_{j=1}^k s_j$. We consider the minimal $\ell \in \mathbb{N}$ such that $\sum_{j=1}^k s_j < 2^\ell$, and we let $d = \ell + k$. We denote by z the zero vector of \mathbb{N}^d , and we denote by e_i the i th unit vector of \mathbb{N}^d defined by $e_i(i) = 1$ and $e_i(j) = 0$ if $j \neq i$. We denote for a natural number $n < 2^\ell$ the vector $\text{bin}(n) \in \mathbb{N}^d$ defined as $\sum_{i=1}^\ell b_i e_i$ where $b_1, \dots, b_\ell \in \{0, 1\}$ are such that $n = \sum_{i=1}^\ell b_i 2^{i-1}$.

We introduce the ultraflat d -VASS $V = (Q, T)$ where Q is the set of states $q_1, p_1, \dots, q_k, p_k, q_{k+1}, \dots, q_{k+\ell}$, and T is the set of transitions that contains the transitions labeled by z that connect the states of Q to form an ultraflat VASS, the transitions $(q_j, -e_{\ell+k}, q_j)$ and $(p_j, -e_{\ell+k} + \text{bin}(s_j), p_j)$ for every $1 \leq j \leq k$, transitions $(q_{k+i}, -2e_i + e_{i+1}, q_{k+i})$ for every $1 \leq i < \ell$.

Just observe that $q_1(\sum_{j=1}^k e_{\ell+j}) \xrightarrow{*} q_{k+\ell}(\text{bin}(s))$ if, and only if, there exists $J \subseteq \{1, \dots, k\}$ such that $s = \sum_{j \in J} s_j$. \square

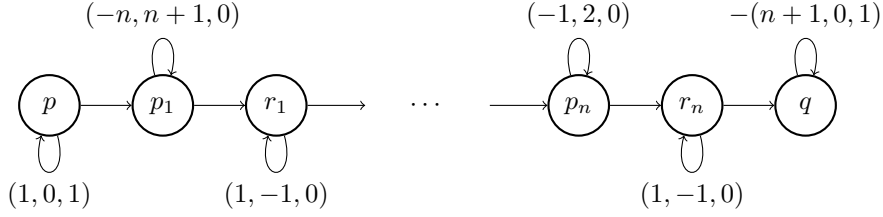
Finally, let us consider the reachability problem for flat d -VASS where d is fixed and numbers are encoded in unary. In this context, the complexity of the problem is difficult to determined since we need to compute with a fix number of counters large numbers with actions that involves only small numbers. This intuition is confirmed up to the dimension 2. In fact, the reachability problem for (not necessarily flat) 1-VASS encoded in unary is NL-complete by using a classical hill-cutting argument. For flat 2-VASS encoded in unary, the reachability problem was also proved to be NL-complete in [23] by observing that if there exists a word σ such that $c_{\text{in}} \xrightarrow{\sigma} c_{\text{out}}$ for a flat 2-VASS, then there exists another one with a length polynomially bounded in the size of the VASS and the configurations $c_{\text{in}}, c_{\text{out}}$ encoded in unary. Such a property is not trivial since an hill-cutting argument can no longer be applied in that context as shown by the following example.

Example 6. Let us introduce the family $(V_n)_{n \in \mathbb{N}}$ of ultraflat 2-VASS parameterized by a natural number n and depicted below (zero vectors are not depicted). Intuitively, iterating the loop on a state p_i transfers and multiplies by two the content of the first counter to the second counter, iterating the loop on a state r_i transfers back the content of the second counter to the first counter, iterating the loop on a state s_i transfers the content of the first counter to the second counter, and iterating the loop on a state q_i transfers back and divides by two the content of the second counter to the first counter. Observe that $p_1(1, 0) \xrightarrow{\sigma_n} q_1(1, 0)$ where σ_n is a run obtained by executing each loop a maximal number of times. It follows that σ_n is exponentially long in n . Moreover, the set C_n of configurations c such that $p_1(1, 0) \xrightarrow{u} c \xrightarrow{v} q_1(1, 0)$ where u, v are such that $\sigma_n = uv$ is an exponential set of incomparable configurations for the relation \sqsubseteq defined by $p(x_1, x_2) \sqsubseteq q(y_1, y_2)$ if $p = q$, $x_1 \leq y_1$ and $x_2 \leq y_2$.



A complexity lowerbound better than NL is open for ultraflat 3-VASS. Starting from that dimension, the minimal length of a word σ such that $c_{\text{in}} \xrightarrow{\sigma} c_{\text{out}}$ can be exponentially long in the size of the VASS and the configurations $c_{\text{in}}, c_{\text{out}}$ encoded in unary as shown by the following example.

Example 7. Let us introduce the family $(V_n)_{n \in \mathbb{N}}$ of ultraflat 3-VASS parameterized by a natural number n and depicted below (zero vectors are not depicted). Those VASS are presented as counter programs in [21]. Intuitively, the loop on state p initialized the VASS from $p(1, 0, 1)$ to a configuration $p_0(x, 0, x)$ where x is any positive natural number. Iterating the loop on a state p_i transfers and multiplies by $\frac{n+2-i}{n+1-i}$ the content of the first counter to the second counter, iterating the loop on a state r_i transfers back the content of the second counter to the first counter. By iterating all those loops, starting from $p_0(x, 0, x)$, if all the multiplications are performed exactly, we get the configuration $q(x(n+1), 0, x)$ since $\prod_{i=1}^n \frac{n+2-i}{n+1-i} = n+1$. From such a configuration, by iterating the loop on state q we get the configuration $q(0, 0, 0)$. In order to obtain such an execution, we prove in [21] that x is necessarily a non zero multiple of $\frac{\text{lcm}(2, \dots, n+1)}{n+1}$ which is exponential in n . It follows that the minimal word σ such that $p(1, 0, 1) \xrightarrow{\sigma} q(0, 0, 0)$ is exponential in n .



Based on Ex. 7, the reachability problem for flat 7-VASS in unary is proved to be NP-complete in [21]. In a work in progress, we recently proved that the problem is also NP-complete in dimension 5. It follows that the best complexity lowerbound of the reachability problem for flat d -VASS is NL for $d \in \{3, 4\}$. Proving that the reachability problem is NP-hard in those dimensions requires to find a way to use counters in a non trivial way. Such a trick, if it exists, could pave the way for increasing the best known complexity lower bound [20] of the general (non flat) reachability problem for VASS in order to match the Ackermannian complexity upper bound [40]. On the other side, the best complexity upperbound of the reachability problem for ultraflat d -VASS encoded in unary is NP for $d \geq 3$. Proving that this problem is in P when $d \in \{3, 4\}$ could provide a way to prove that the reachability problem for flat d -VASS in unary is in P.

5 Conclusion

We presented in this paper an overview of results about the reachability problem for VASS related to Presburger arithmetic. Those results came from several peer reviewed publications, except Lem. 1 and Lem. 2 (we are confident that those results are correct, but, in case of, notice that when replacing ultraflat by flat in those lemmas, proofs are almost immediate).

The complexity of the reachability problem for general VASS is still open. The best known complexity lowerbound is a tower function with an height limited by the dimension of the VASS. A possible way to lift up this bound to match the Ackermannian upperbound is to find a trick to reuse some counters in a non trivial way. We think that looking at the complexity of flat 3-VASS encoded in unary may pave the way to get such a trick since if this problem is NP-hard, in order to prove such a bound, we need to encode an NP-hard problem like the subsetsum problem with only three counters and small actions. Intuitively some counters must be reused along the computation. On the other hand, if the problem is in P, we may obtain from the proof of that result a way to prove that the reachability problem for general 3-VASS is elementary (the best known upperbound is tower for structurally bounded 3-VASS).

In this paper, we did not present the vast set of results related to Petri net extensions. Anyway, let us mention that the complexity of the reachability problem for flat 3-VASS encoded in unary seems to be related to the complexity of the coverability problem for pushdown 1-VASS [22].

Bibliography

- [1] Angluin, D., Aspnes, J., Diamadi, Z., Fischer, M.J., Peralta, R.: Computation in networks of passively mobile finite-state sensors. In: Chaudhuri, S., Kutten, S. (eds.) Proceedings of the Twenty-Third Annual ACM Symposium on Principles of Distributed Computing, PODC 2004, St. John's, Newfoundland, Canada, July 25-28, 2004. pp. 290–299. ACM (2004). <https://doi.org/10.1145/1011767.1011810>, <https://doi.org/10.1145/1011767.1011810>
- [2] Annichini, A., Asarin, E., Bouajjani, A.: Symbolic techniques for parametric reasoning about counter and clock systems. In: Emerson, E.A., Sistla, A.P. (eds.) Computer Aided Verification, 12th International Conference, CAV 2000, Chicago, IL, USA, July 15-19, 2000, Proceedings. Lecture Notes in Computer Science, vol. 1855, pp. 419–434. Springer (2000). https://doi.org/10.1007/10722167_32, https://doi.org/10.1007/10722167_32
- [3] Annichini, A., Bouajjani, A., Sighireanu, M.: Trex: A tool for reachability analysis of complex systems. In: Berry, G., Comon, H., Finkel, A. (eds.) Computer Aided Verification, 13th International Conference, CAV 2001, Paris, France, July 18-22, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2102, pp. 368–372. Springer (2001). https://doi.org/10.1007/3-540-44585-4_34, https://doi.org/10.1007/3-540-44585-4_34
- [4] Bardin, S., Finkel, A., Leroux, J.: Faster acceleration of counter automata in practice. In: Tools and Algorithms for the Construction and Analysis of Systems, 10th International Conference, TACAS 2004, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2004, Barcelona, Spain, March 29 - April 2, 2004, Proceedings. Lecture Notes in Computer Science, vol. 2988, pp. 576–590. Springer (2004). https://doi.org/10.1007/978-3-540-24730-2_42
- [5] Bardin, S., Finkel, A., Leroux, J., Petrucci, L.: FAST: fast acceleration of symbolike transition systems. In: Computer Aided Verification, 15th International Conference, CAV 2003, Boulder, CO, USA, July 8-12, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2725, pp. 118–121. Springer (2003). https://doi.org/10.1007/978-3-540-45069-6_12
- [6] Bardin, S., Leroux, J., Point, G.: FAST extended release. In: Computer Aided Verification, 18th International Conference, CAV 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4144, pp. 63–66. Springer (2006). https://doi.org/10.1007/11817963_9
- [7] Bardin, S., Finkel, A., Leroux, J., Schnoebelen, P.: Flat acceleration in symbolic model checking. In: Peled, D.A., Tsay, Y. (eds.) Automated Technology for Verification and Analysis, Third International Symposium, ATVA 2005, Taipei, Taiwan, October 4-7, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3707, pp. 474–488. Springer

- (2005). https://doi.org/10.1007/11562948_35, https://doi.org/10.1007/11562948_35
- [8] Boigelot, B.: On iterating linear transformations over recognizable sets of integers. *Theor. Comput. Sci.* **309**(1-3), 413–468 (2003). [https://doi.org/10.1016/S0304-3975\(03\)00314-1](https://doi.org/10.1016/S0304-3975(03)00314-1), [https://doi.org/10.1016/S0304-3975\(03\)00314-1](https://doi.org/10.1016/S0304-3975(03)00314-1)
- [9] Boigelot, B.: Domain-specific regular acceleration. *Int. J. Softw. Tools Technol. Transf.* **14**(2), 193–206 (2012). <https://doi.org/10.1007/s10009-011-0206-x>, <https://doi.org/10.1007/s10009-011-0206-x>
- [10] Boigelot, B., Godefroid, P.: Symbolic verification of communication protocols with infinite state spaces using qdds. *Formal Methods Syst. Des.* **14**(3), 237–255 (1999). <https://doi.org/10.1023/A:1008719024240>, <https://doi.org/10.1023/A:1008719024240>
- [11] Boigelot, B., Godefroid, P., Willems, B., Wolper, P.: The power of qdds (extended abstract). In: Hentenryck, P.V. (ed.) *Static Analysis*, 4th International Symposium, SAS '97, Paris, France, September 8-10, 1997, Proceedings. *Lecture Notes in Computer Science*, vol. 1302, pp. 172–186. Springer (1997). <https://doi.org/10.1007/BFb0032741>, <https://doi.org/10.1007/BFb0032741>
- [12] Boigelot, B., Herbreteau, F.: The power of hybrid acceleration. In: Ball, T., Jones, R.B. (eds.) *Computer Aided Verification*, 18th International Conference, CAV 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings. *Lecture Notes in Computer Science*, vol. 4144, pp. 438–451. Springer (2006). https://doi.org/10.1007/11817963_40, https://doi.org/10.1007/11817963_40
- [13] Boigelot, B., Herbreteau, F., Mainz, I.: Acceleration of affine hybrid transformations. In: Cassez, F., Raskin, J. (eds.) *Automated Technology for Verification and Analysis - 12th International Symposium, ATVA 2014*, Sydney, NSW, Australia, November 3-7, 2014, Proceedings. *Lecture Notes in Computer Science*, vol. 8837, pp. 31–46. Springer (2014). https://doi.org/10.1007/978-3-319-11936-6_4, https://doi.org/10.1007/978-3-319-11936-6_4
- [14] Boigelot, B., Wolper, P.: Symbolic verification with periodic sets. In: Dill, D.L. (ed.) *Computer Aided Verification*, 6th International Conference, CAV '94, Stanford, California, USA, June 21-23, 1994, Proceedings. *Lecture Notes in Computer Science*, vol. 818, pp. 55–67. Springer (1994). https://doi.org/10.1007/3-540-58179-0_43, https://doi.org/10.1007/3-540-58179-0_43
- [15] Bouajjani, A., Habermehl, P.: Symbolic reachability analysis of FIFO channel systems with nonregular sets of configurations (extended abstract). In: Degano, P., Gorrieri, R., Marchetti-Spaccamela, A. (eds.) *Automata, Languages and Programming*, 24th International Colloquium, ICALP'97, Bologna, Italy, 7-11 July 1997, Proceedings. *Lecture Notes in Computer Science*, vol. 1256, pp. 560–570. Springer (1997). https://doi.org/10.1007/3-540-63165-8_211, https://doi.org/10.1007/3-540-63165-8_211

- [16] Bouajjani, A., Habermehl, P.: Symbolic reachability analysis of fifo-channel systems with nonregular sets of configurations. *Theor. Comput. Sci.* **221**(1-2), 211–250 (1999). [https://doi.org/10.1016/S0304-3975\(99\)00033-X](https://doi.org/10.1016/S0304-3975(99)00033-X), [https://doi.org/10.1016/S0304-3975\(99\)00033-X](https://doi.org/10.1016/S0304-3975(99)00033-X)
- [17] Bozga, M., Iosif, R.: On flat programs with lists. In: Cook, B., Podelski, A. (eds.) *Verification, Model Checking, and Abstract Interpretation, 8th International Conference, VMCAI 2007, Nice, France, January 14–16, 2007, Proceedings. Lecture Notes in Computer Science*, vol. 4349, pp. 122–136. Springer (2007). https://doi.org/10.1007/978-3-540-69738-1_9, https://doi.org/10.1007/978-3-540-69738-1_9
- [18] Bozga, M., Iosif, R., Lakhnech, Y.: Flat parametric counter automata. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10–14, 2006, Proceedings, Part II. Lecture Notes in Computer Science*, vol. 4052, pp. 577–588. Springer (2006). https://doi.org/10.1007/11787006_49, https://doi.org/10.1007/11787006_49
- [19] Bozga, M., Iosif, R., Lakhnech, Y.: Flat parametric counter automata. *Fundam. Informaticae* **91**(2), 275–303 (2009). <https://doi.org/10.3233/FI-2009-0044>, <https://doi.org/10.3233/FI-2009-0044>
- [20] Czerwiński, W., Lasota, S., Lazić, R., Leroux, J., Mazowiecki, F.: The reachability problem for petri nets is not elementary. In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23–26, 2019*. pp. 24–33. ACM (2019). <https://doi.org/10.1145/3313276.3316369>
- [21] Czerwinski, W., Lasota, S., Lazic, R., Leroux, J., Mazowiecki, F.: Reachability in fixed dimension vector addition systems with states. In: Konnov, I., Kovács, L. (eds.) *31st International Conference on Concurrency Theory, CONCUR 2020, September 1–4, 2020, Vienna, Austria (Virtual Conference). LIPIcs*, vol. 171, pp. 48:1–48:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020). <https://doi.org/10.4230/LIPIcs.CONCUR.2020.48>, <https://doi.org/10.4230/LIPIcs.CONCUR.2020.48>
- [22] Englert, M., Hofman, P., Lasota, S., Lazic, R., Leroux, J., Straszynski, J.: A lower bound for the coverability problem in acyclic pushdown VAS. *Inf. Process. Lett.* **167**, 106079 (2021). <https://doi.org/10.1016/j.ipl.2020.106079>, <https://doi.org/10.1016/j.ipl.2020.106079>
- [23] Englert, M., Lazic, R., Totzke, P.: Reachability in two-dimensional unary vector addition systems with states is nl -complete. In: Grohe, M., Koskinen, E., Shankar, N. (eds.) *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5–8, 2016*. pp. 477–484. ACM (2016). <https://doi.org/10.1145/2933575.2933577>, <https://doi.org/10.1145/2933575.2933577>
- [24] Esparza, J., Nielsen, M.: Decidability issues for petri nets - a survey. *Bulletin of the European Association for Theoretical Computer Science* **52**, 245–262 (1994)

- [25] Esparza, J., Ganty, P., Leroux, J., Majumdar, R.: Verification of population protocols. In: Aceto, L., de Frutos-Escrig, D. (eds.) 26th International Conference on Concurrency Theory, CONCUR 2015, Madrid, Spain, September 1-4, 2015. LIPIcs, vol. 42, pp. 470–482. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2015). <https://doi.org/10.4230/LIPIcs.CONCUR.2015.470>, <https://doi.org/10.4230/LIPIcs.CONCUR.2015.470>
- [26] Finkel, A., Leroux, J.: How to compose presburger-accelerations: Applications to broadcast protocols. In: FST TCS 2002: Foundations of Software Technology and Theoretical Computer Science, 22nd Conference Kanpur, India, December 12-14, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2556, pp. 145–156. Springer (2002). https://doi.org/10.1007/3-540-36206-1_14
- [27] Fribourg, L., Olsén, H.: Proving safety properties of infinite state systems by compilation into presburger arithmetic. In: Mazurkiewicz, A.W., Winkowski, J. (eds.) CONCUR '97: Concurrency Theory, 8th International Conference, Warsaw, Poland, July 1-4, 1997, Proceedings. Lecture Notes in Computer Science, vol. 1243, pp. 213–227. Springer (1997). https://doi.org/10.1007/3-540-63141-0_15, https://doi.org/10.1007/3-540-63141-0_15
- [28] Hack, M.H.T.: Decidability questions for Petri nets. Ph.D. thesis, MIT (1975), <http://publications.csail.mit.edu/lcs/pubs/pdf/MIT-LCS-TR-161.pdf>
- [29] Hauschildt, D.: Semilinearity of the Reachability Set is Decidable for Petri Nets. Ph.D. thesis, University of Hamburg (1990)
- [30] Hopcroft, J.E., Pansiot, J.J.: On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science* **8**, 135–159 (1979)
- [31] Karp, R.M., Miller, R.E.: Parallel program schemata. *J. Comput. Syst. Sci.* **3**(2), 147–195 (1969). [https://doi.org/10.1016/S0022-0000\(69\)80011-5](https://doi.org/10.1016/S0022-0000(69)80011-5)
- [32] Kosaraju, S.R.: Decidability of reachability in vector addition systems (preliminary version). In: STOC. pp. 267–281. ACM (1982). <https://doi.org/10.1145/800070.802201>
- [33] Kosaraju, S.R., Sullivan, G.F.: Detecting cycles in dynamic graphs in polynomial time (preliminary version). In: Simon, J. (ed.) Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA. pp. 398–406. ACM (1988). <https://doi.org/10.1145/62212.62251>, <http://doi.acm.org/10.1145/62212.62251>
- [34] Lambert, J.: A structure to decide reachability in Petri nets. *Theor. Comput. Sci.* **99**(1), 79–104 (1992). [https://doi.org/10.1016/0304-3975\(92\)90173-D](https://doi.org/10.1016/0304-3975(92)90173-D)
- [35] Leroux, J.: The general vector addition system reachability problem by Presburger inductive invariants. *Logical Methods in Computer Science* **6**(3) (2010). [https://doi.org/10.2168/LMCS-6\(3:2\)2010](https://doi.org/10.2168/LMCS-6(3:2)2010)
- [36] Leroux, J.: Vector addition system reachability problem: a short self-contained proof. In: POPL. pp. 307–316. ACM (2011). <https://doi.org/10.1145/1926385.1926421>

- [37] Leroux, J.: Vector addition systems reachability problem (A simpler solution). In: Turing-100. EPiC Series in Computing, vol. 10, pp. 214–228. EasyChair (2012), <http://www.easychair.org/publications/paper/106497>
- [38] Leroux, J.: Presburger vector addition systems. In: 28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013. pp. 23–32. IEEE Computer Society (2013). <https://doi.org/10.1109/LICS.2013.7>
- [39] Leroux, J., Schmitz, S.: Demystifying reachability in vector addition systems. In: 30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015. pp. 56–67. IEEE Computer Society (2015). <https://doi.org/10.1109/LICS.2015.16>
- [40] Leroux, J., Schmitz, S.: Reachability in vector addition systems is primitive-recursive in fixed dimension. In: 34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019. pp. 1–13. IEEE (2019). <https://doi.org/10.1109/LICS.2019.8785796>
- [41] Leroux, J., Sutre, G.: On flatness for 2-dimensional vector addition systems with states. In: CONCUR 2004 - Concurrency Theory, 15th International Conference, London, UK, August 31 - September 3, 2004, Proceedings. Lecture Notes in Computer Science, vol. 3170, pp. 402–416. Springer (2004). https://doi.org/10.1007/978-3-540-28644-8_26
- [42] Leroux, J., Sutre, G.: Flat counter automata almost everywhere! In: Software Verification: Infinite-State Model Checking and Static Program Analysis, 19.02. - 24.02.2006. Dagstuhl Seminar Proceedings, vol. 06081. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany (2006), <http://drops.dagstuhl.de/opus/volltexte/2006/729>
- [43] Leroux, J.: When reachability meets grzegorzcyk. In: Hermanns, H., Zhang, L., Kobayashi, N., Miller, D. (eds.) LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020. pp. 1–6. ACM (2020). <https://doi.org/10.1145/3373718.3394732>, <https://doi.org/10.1145/3373718.3394732>
- [44] Mayr, E.W.: An algorithm for the general petri net reachability problem. In: Proceedings of the 13th Annual ACM Symposium on Theory of Computing, May 11-13, 1981, Milwaukee, Wisconsin, USA. pp. 238–246. ACM (1981). <https://doi.org/10.1145/800076.802477>
- [45] Mayr, E.W.: An algorithm for the general Petri net reachability problem. *SIAM J. Comput.* **13**(3), 441–460 (1984). <https://doi.org/10.1137/0213029>
- [46] Mayr, E.W., Meyer, A.R.: The complexity of the finite containment problem for petri nets. *J. ACM* **28**(3), 561–576 (1981). <https://doi.org/10.1145/322261.322271>
- [47] McAloon, K.: Petri nets and large finite sets. *Theor. Comput. Sci.* **32**, 173–183 (1984). [https://doi.org/10.1016/0304-3975\(84\)90029-X](https://doi.org/10.1016/0304-3975(84)90029-X)
- [48] Sacerdote, G.S., Tenney, R.L.: The decidability of the reachability problem for vector addition systems (preliminary version). In: Proceedings of the 9th Annual ACM Symposium on Theory of Comput-

ing, May 4-6, 1977, Boulder, Colorado, USA. pp. 61–76. ACM (1977).
<https://doi.org/10.1145/800105.803396>

- [49] Schmitz, S.: The complexity of reachability in vector addition systems. SIGLOG News **3**(1), 4–21 (2016), <https://dl.acm.org/citation.cfm?id=2893585>