Improved Algorithm for Reachability in d-VASS

Yuxi Fu
BASICS, Shanghai Jiao Tong University, China

Qizhe Yang[1]
Shanghai Normal University, China

Yangluo Zheng
BASICS, Shanghai Jiao Tong University, China

Abstract

An F_d upper bound for the reachability problem in vector addition systems with states (VASS) in fixed dimension is given, where F_d is the d-th level of the Grzegorczyk hierarchy of complexity classes. The new algorithm combines the idea of the linear path scheme characterization of the reachability in the 2-dimension VASSes with the general decomposition algorithm by Mayr, Kosaraju and Lambert. The result improves the F_{d+4} upper bound due to Leroux and Schmitz (LICS 2019).

2012 ACM Subject Classification Theory of computation → Models of computation; Theory of computation → Logic and verification

Keywords and phrases Petri net, vector addition system, reachability

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.136

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Related Version *Full Version*: https://arxiv.org/abs/2404.14781

Funding The support from the National Natural Science Foundation of China (62072299) is acknowledged.

Acknowledgements We thank Weijun Chen, Huan Long, Hao Wu and Qiang Yin for proofreading various versions of this paper.

1 Introduction

Petri nets, or equivalently vector addition system with states (VASS), are a well studied model of concurrency. A VASS consists of a finite state control where each state transition has as its effect an integer-valued vector, and its configurations are pairs of a state and a vector with *natural number* components. A transition may lead from one configuration to another by adding its effect component-wise, conditioned on that the components of the resulting vector remain non-negative. The *reachability problem*, which asks whether from one configuration there is a path reaching another configuration, lies in the center of the algorithmic theory of Petri nets and has found a wide range of applications due to its generic nature. Since the problem was shown to be decidable by Mayr [17] in 1981, its computational complexity had been a long-standing open problem in the field. In 2015, Leroux and Schmitz [14] presented the first complexity upper bound, stating that the reachability problem is cubic-Ackermannian. This was later improved to an Ackermannian upper bound in 2019, again by Leroux and Schmitz [15]. Regarding the hardness, in 2021 seminal works by Czerwiński and Orlikowski [6], and independently by Leroux [13], provided matching Ackermannian lower bounds, settling the exact complexity of the problem.

[1] corresponding author

51st International Colloquium on Automata, Languages, and Programming (ICALP 2024).
Editors: Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson;
Article No. 136; pp. 136:1–136:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Concerning the parameterization by dimension, i.e. the reachability problem in d-dimensional VASSes where d is fixed, there is still a gap in the known complexity bounds. Currently, we only have exact complexity bounds for dimension one and two [7, 1]. For dimension $d \geq 3$, the result of Leroux and Schmitz [15] shows that the problem is in F_{d+4}, the (d+4)-th level of the Grzegorczyk hierarchy of complexity classes. Note that a recent work by Yang and Fu [22] points out that the problem for dimension 3 is in $\mathsf{F}_3 = \mathsf{TOWER}$. On the other hand, the best known lower bound by Czerwiński, Jecker, Lasota, Leroux and Orlikowski [5] states that reachability in (2d+3)-dimensional VASSes is F_d-hard. Motivated by this gap, our paper focuses on the computational complexity of reachability problem in the fixed-dimensional VASSes.

Our contribution

In this paper we show that the reachability problem in the d-dimensional VASS is in F_d for $d \geq 3$, improving the previous F_{d+4} upper bound by Leroux and Schmitz [15], and generalizing the tower upper bound for the reachability problem in 3-VASS [22]. The new upper bound is obtained with the help of two novel technical lemmas.

1. Our main technical tool (Theorem 3.4) is a generalization of the linear path scheme characterization for the reachability relation in the 2-dimensional VASSes [1]. By borrowing the key idea from the work of Yang and Fu [22], we show that as long as the "geometric dimension" of a VASS (that is, the dimension of the vector space spanned by the effects of cyclic paths) is bounded by 2, its reachability relation can be characterized by short linear path schemes. We then apply the lemma to simplify the KLMST algorithm so that (i) a VASS is replaced by a short linear path scheme whenever its geometric dimension is no more than 2 and (ii) the linear path schemes will not be decomposed further. It is then routine [15], using the tools from [21], to show that the reachability problem in the d-dimensional VASS is in F_{d+1} for all $d \geq 3$.
2. Our second lemma (Lemma 6.3) allows us to improve further the bound from F_{d+1} to F_d. This is done by a careful analysis of the properties of the fast-growing functions [21].

Due to space limitation the proofs of the two lemmas are placed in the appendices.

Organization

Section 2 fixes notation, defines the VASS model and its reachability problem. Section 3 generalizes the linear path scheme characterization [1] to VASSes whose geometric dimension are bounded by 2. Section 4 recalls the characterization system of linear inequalities for linear path schemes. Section 5 makes use of the results of Section 3 to give an improved version of the classic KLMST decomposition algorithm. Section 6 analyzes the complexity of our modified algorithm, proving the main result. Section 7 concludes. Proofs omitted from the main text can be found in the appendices.

2 Preliminaries

We use $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$ to denote respectively the set of non-negative integers, integers, and rational numbers. Let $n \in \mathbb{N}$ be a number, we write $[n]$ for the range $\{1, 2, \ldots, n\}$. Let $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{X}^d$ be d-dimensional vectors where $\mathbb{X}$ can be any one of $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$. We write $\boldsymbol{v}(i)$ for the i-th component of $\boldsymbol{v}$ where $i \in [d]$, so $\boldsymbol{v} = (\boldsymbol{v}(1), \ldots, \boldsymbol{v}(d))$. The *maximum norm* of $\boldsymbol{v}$ is defined to be $\|\boldsymbol{v}\| := \max_{i \in [d]} |\boldsymbol{v}(i)|$. We extend component-wise the order $\leq$ for vectors in $\mathbb{X}^d$, so $\boldsymbol{u} \leq \boldsymbol{v}$ if and only if $\boldsymbol{u}(i) \leq \boldsymbol{v}(i)$ for all $i \in [d]$. Addition and subtraction of vectors are also component-wise, so $(\boldsymbol{u}+\boldsymbol{v})(i) = \boldsymbol{u}(i) + \boldsymbol{v}(i)$ for all $i \in [d]$. Define $\text{supp}(\boldsymbol{v}) := \{i \in [d] : \boldsymbol{v}(i) \neq 0\}$ to be the set of indices of non-zero components of $\boldsymbol{v}$. This notation is extended to sets of vectors naturally, so $\text{supp}(S) = \bigcup_{\boldsymbol{v}\in S} \text{supp}(\boldsymbol{v})$ for any set $S \subseteq \mathbb{X}^d$.

For technical reasons we introduce the symbol ω that stands for the infinite element. Let $\mathbb{N}_\omega := \mathbb{N} \cup \{\omega\}$. We stipulate that $n < \omega$ for all $n \in \mathbb{N}$, and $x + \omega = \omega + x = \omega$ for all $x \in \mathbb{Z}$. Define the partial order $\sqsubseteq$ over $\mathbb{N}_\omega$ so that $x \sqsubseteq y$ if and only if $x = y$ or $y = \omega$ for all $x, y \in \mathbb{N}_\omega$. The relation $\sqsubseteq$ is also extended component-wise to vectors in $\mathbb{N}_\omega^d$.

Let Σ be a finite alphabet and $s, t \in \Sigma^*$ be two strings over Σ. We write st for the concatenation of s and t, and s^n for the concatenation of n copies of s where $n \in \mathbb{N}$. If $s = a_1 a_2 \ldots a_\ell$ where $a_1, \ldots, a_\ell \in \Sigma$, we write $|s| := \ell$ for the length of s, and $s[i \ldots j] := a_i a_{i+1} \ldots a_j$ for the substring of s between indices i and j where $1 \leq i \leq j \leq |s|$.

2.1 Vector Addition Systems with States

Let $d \geq 0$ be an integer. A *d-dimensional vector addition system with states (d-VASS)* is a pair $G = (Q, T)$ where Q is a finite set of *states* and $T \subseteq Q \times \mathbb{Z}^d \times Q$ is a finite set of *transitions*. Clearly a VASS can also be viewed as a directed graph with edges labelled by integer vectors. Given a word $\pi = (p_1, \boldsymbol{a}_1, q_1)(p_2, \boldsymbol{a}_2, q_2) \ldots (p_n, \boldsymbol{a}_n, q_n) \in T^*$ over transitions, we say that π is a *path from p_1 to q_n* if $q_i = p_{i+1}$ for all $i = 1, \ldots, n-1$. It is a *cycle* if we further have $p_1 = q_n$. The *effect* of π is defined to be $\Delta(\pi) := \sum_{i=1}^{n} \boldsymbol{a}_i$, and the *action word* of π is the word $[\![\pi]\!] := \boldsymbol{a}_1 \boldsymbol{a}_2 \ldots \boldsymbol{a}_n$ over $\mathbb{Z}^d$. The *Parikh image* of π is the function $\phi \in \mathbb{N}^T$ mapping each transition to its number of occurrences in π. Given a function $\phi \in \mathbb{N}^T$ we also define $\Delta(\phi) := \sum_{t=(p,\boldsymbol{a},q)\in T} \phi(t) \cdot \boldsymbol{a}$. Note that $\Delta(\phi) = \Delta(\pi)$ if ϕ is the Parikh image of π. Let $L \subseteq T^*$ be a language (i.e. subset of words), we define its effect as $\Delta(L) := \{\Delta(\pi) : \pi \in L\}$.

The norm of a transition $t = (p, \boldsymbol{a}, q)$ is defined by $\|t\| := \|\boldsymbol{a}\|$. The norm of a path $\pi = t_1 t_2 \ldots t_n$ is $\|\pi\| := \max_{i\in[n]} \|t_i\|$. For a d-VASS $G = (Q, T)$ we write $\|T\| := \max\{\|t\| : t \in T\}$. The *size* of G is defined by

$$|G| := |Q| + |T| + d \cdot |T| \cdot \|T\| \,. \tag{1}$$

Semantics of VASSes

Let $G = (Q, T)$ be a d-VASS. A *configuration* of G is a pair of a state $p \in Q$ and a vector $\boldsymbol{v} \in \mathbb{Z}^d$, written as $p(\boldsymbol{v})$. Let $\mathbb{D} \subseteq \mathbb{Z}^d$, we define the $\mathbb{D}$-semantics for G as follows. For each transition $t = (p, \boldsymbol{a}, q) \in T$, the one-step transition relation $\xrightarrow{t}_{\mathbb{D}}$ relates all pairs of configurations of the form $(p(\boldsymbol{u}), q(\boldsymbol{v}))$ where $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{D}$ and $\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{a}$. Then for a word $\pi = t_1 t_2 \ldots t_n \in T^*$, the relation $\xrightarrow{\pi}_{\mathbb{D}}$ is the composition $\xrightarrow{\pi}_{\mathbb{D}} \; := \; \xrightarrow{t_1}_{\mathbb{D}} \circ \cdots \circ \xrightarrow{t_n}_{\mathbb{D}}$. So $p(\boldsymbol{u}) \xrightarrow{\pi}_{\mathbb{D}} q(\boldsymbol{v})$ if and only if there are configurations $p_0(\boldsymbol{u}_0), \ldots, p_n(\boldsymbol{u}_n) \in Q \times \mathbb{D}$ such that

$$p(\boldsymbol{u}) = p_0(\boldsymbol{u}_0) \xrightarrow{t_1}_{\mathbb{D}} p_1(\boldsymbol{u}_1) \xrightarrow{t_2}_{\mathbb{D}} \cdots \xrightarrow{t_n}_{\mathbb{D}} p_n(\boldsymbol{u}_n) = q(\boldsymbol{v}). \tag{2}$$

Also, when $\pi = \epsilon$ is the empty word, the relation $\xrightarrow{\epsilon}_{\mathbb{D}}$ is the identity relation over $Q \times \mathbb{D}$. Note that $\xrightarrow{\pi}_{\mathbb{D}}$ is non-empty only if π is a path. When $p(\boldsymbol{u}) \xrightarrow{\pi}_{\mathbb{D}} q(\boldsymbol{v})$ we also say that π induces (or is) a $\mathbb{D}$-*run* from $p(\boldsymbol{u})$ to $q(\boldsymbol{v})$. We emphasize that all configurations on this run lie in $\mathbb{D}$, and that they are uniquely determined by $p(\boldsymbol{u})$ and π. For a language $L \subseteq T^*$ we define $\xrightarrow{L}_{\mathbb{D}}$ as $\bigcup_{\pi\in L} \xrightarrow{\pi}_{\mathbb{D}}$. Finally, the $\mathbb{D}$-*reachability relation* of G is defined to be $\xrightarrow{*}_{\mathbb{D}} \; := \; \xrightarrow{T^*}_{\mathbb{D}}$.

For the above definitions, we shall often omit the subscript $\mathbb{D}$ when $\mathbb{D} = \mathbb{N}^d$.

We mention that in Section 5 we need to generalize the VASS semantics to configurations in $Q \times \mathbb{N}_\omega^d$, allowing ω components in vectors. The definitions of $\xrightarrow{t}_{\mathbb{N}_\omega^d}$, $\xrightarrow{\pi}_{\mathbb{N}_\omega^d}$, and $\xrightarrow{*}_{\mathbb{N}_\omega^d}$ are similar to the above.

Reachability problem

The reachability problem in vector addition systems with states is formulated as follows:

REACHABILITY IN d-DIMENSIONAL VECTOR ADDITION SYSTEM WITH STATES
Input: A d-dimensional VASS $G = (Q, T)$, two configurations $p(\boldsymbol{u}), q(\boldsymbol{v}) \in Q \times \mathbb{N}^d$.
Question: Does $p(\boldsymbol{u}) \xrightarrow{*}_{\mathbb{N}^d} q(\boldsymbol{v})$ hold in G?

Note that we study the reachability problem for fixed-dimensional VASSes, where the dimension d is treated as a constant to allow more fine-grained analysis. So we shall use the big-O notation to hide constants that may depend on d. The general problem where the dimension can be part of the input was already shown to be Ackermann-complete [6, 13].

Cycle spaces and geometric dimensions

One of the key insights of [15] is a new termination argument for the KLMST decomposition algorithm based on the dimensions of vector spaces spanned by cycles in VASSes, which yielded the primitive recursive upper bound of VASS reachability problem in fixed dimensions. The vector spaces spanned by cycles also play an important role in our work.

▶ **Definition 2.1.** *Let G be a d-VASS. The* cycle space *of G is the vector space* $\mathrm{Cyc}(G) \subseteq \mathbb{Q}^d$ *spanned by the effects of all cycles in G, that is:*

$$\mathrm{Cyc}(G) := \mathrm{span}\{\Delta(\beta) : \beta \textit{ is a cycle in } G\}. \tag{3}$$

The dimension of the cycle space of G is called the geometric dimension *of G. We say G is* geometrically k-dimensional *if* $\dim(\mathrm{Cyc}(G)) \leq k$.

3 Flattability of Geometrically 2-dimensional VASSes

A VASS is *flat* if each of its states lies on at most one cycle. Flat VASSes form an important subclass of VASSes due to its connection to Presburger arithmetic, and we refer the readers to [12] for a survey. In dimension 2, it was proved in [1] that 2-VASSes enjoy a stronger form of flat representation, known as the *linear path schemes*. A linear path scheme is a regular expression of the form $\alpha_0\beta_1^*\alpha_1 \ldots \beta_k^*\alpha_k$ where $\alpha_0, \ldots \alpha_k$ are paths and $\beta_1, \ldots, \beta_k$ are cycles of the VASS, such that they form a path when joined together. The results of [1] show that the reachability relation of every 2-VASS can be captured by short linear path schemes.

Linear path schemes are extremely useful as they can be fully characterized by linear inequality systems so that standard tools for linear or integer programming can be applied. In this section, we generalize the results in [1] and show that the reachability relation of any d-VASS whose geometric dimension is bounded by 2 can also be captured by short linear path schemes.

Our proof follows closely to the lines of [1]. Given a geometrically 2-dimensional VASS G, we first cover $\mathbb{N}^d$ by the following two regions: one for the region far away from every axis:

$$\mathbb{O} := \{\boldsymbol{u} \in \mathbb{N}^d : \boldsymbol{u}(i) \geq D \text{ for all } i \in [d]\} \tag{4}$$

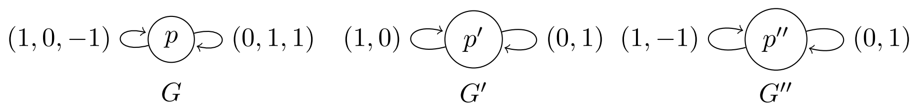

Figure 1 A geometrically 2-dimensional VASS G and two possible projections of it.

where D is some properly chosen threshold; the other one for the region close to some axis:

$$\mathbb{L} := \{\boldsymbol{u} \in \mathbb{N}^d : \boldsymbol{u}(i) \le D' \text{ for some } i \in [d]\} \tag{5}$$

where D' is chosen slightly above D to create an overlap with $\mathbb{O}$. Let π be a run that we are going to capture by linear path schemes. We can extract its maximal prefix that lies completely in either $\mathbb{O}$ or $\mathbb{L}$, depending on where π starts. This prefix must end at a configuration that lies in $\mathbb{L} \cap \mathbb{O}$, if we haven't touched the end of π. From this configuration we then extract a maximal cycle that also ends in $\mathbb{L} \cap \mathbb{O}$. Continuing this fashion, we can break π into segments of runs that lie completely in $\mathbb{O}$ or $\mathbb{L}$, interleaved by cycles that start and end in $\mathbb{O}$ (actually in $\mathbb{L} \cap \mathbb{O}$). Note that the number of such cycles cannot exceed the number of states of G since they are maximal. Now we only need to capture the following three types of runs by short linear path schemes:

1. Runs that are cycles starting and ending in $\mathbb{O}$.
   This will be handled in Section A.4 in the appendix. Briefly speaking, since the geometric dimension of G is 2, the effect of such a cycle must belong to a plane in $\mathbb{Z}^d$. We will find a clever way to project this plane to a coordinate plane, and then project the d-VASS G onto this plane to get a 2-VASS. This is made possible by a novel technique called the "sign-reflecting projection" developed in Section A.3. Intuitively speaking, for any vector in a plane we are able to determine whether it belongs to a certain orthant by observing only 2 entries of this vector. The d-VASS can then be projected onto these 2 coordinates. (See Example 3.1 for a more concrete demonstration.) Now we apply the results of [1] to obtain a linear path scheme that captures the projected cyclic run. Combined with a lemma in [16], the projection guarantees that we can safely project it back to get a linear path scheme for the run in G.
2. Runs that lie completely in $\mathbb{O}$.
   This is just an easy corollary of the first type, and will also be handled in Section A.4. Just note that any run can be broken into a series of simple paths interleaved by cycles.
3. Runs that lie completely in $\mathbb{L}$.
   This will be handled in Section A.5, by a long and tedious case analysis. In principle, we are going to argue that any such run essentially corresponds to a run in some $(d-1)$-VASS, so that we can use induction.

▶ **Example 3.1.** Consider the geometrically 2-dimensional 3-VASS G as shown in Figure 1. It consists of a single state p and two transitions with effects $(1,0,-1)$ and $(0,1,1)$. In order to apply the results of [1], one would like to derive a 2-VASS from G that reflects runs in G. A simple idea is to discard one coordinate of G. Two possibilities of this idea are shown in Figure 1 as G', which discards the third coordinate, and G'', which discards the second coordinate. However, not all of them are satisfactory. For example, the legal run $p(0,0) \xrightarrow{(1,0)} p(1,0)$ in G' reflects an illegal run $p(0,0,0) \xrightarrow{(1,0,-1)} p(1,0,-1)$ in G where the third coordinate goes negative. On the other hand, all runs in G'' can be safely projected back to a run in G. To see this, just observe that for any vector $\boldsymbol{v}$ in the linear span of $(1,0,-1)$ and $(0,1,1)$, $\boldsymbol{v} \ge \boldsymbol{0}$ if and only if $\boldsymbol{v}(1) \ge 0$ and $\boldsymbol{v}(3) \ge 0$. Thus we can safely discard the second coordinate.

In general, the "sign-reflecting projection" developed in Section A.3 shows that any geometrically 2-dimensional VASS can be projected onto two coordinates so that the signs of these two coordinates reflects the signs of other coordinates.

In the rest of this section we just state formally our main technical results. The detailed proofs are placed in the appendix.

3.1 Linear Path Schemes

A *linear path scheme (LPS)* is a pair (G, Λ) where G is a VASS and Λ is a regular expression of the form $\Lambda = \alpha_0\beta_1^*\alpha_1 \dots \beta_k^*\alpha_k$ such that $\alpha_0, \dots, \alpha_k$ are paths in G and $\beta_1, \dots, \beta_k$ are cycles in G, and $\alpha_0\beta_1\alpha_1 \dots \beta_k\alpha_k$ is a path in G. We say an LPS (G, Λ) is *compatible to* a VASS G' if G' contains all states and transitions in Λ. Very often we shall omit the VASS G and say that Λ on its own is an LPS, with G understood as any VASS to which Λ is compatible. We write $|\Lambda| = |\alpha_0\beta_1\alpha_1 \dots \beta_k\alpha_k|$ for the length of Λ, $\|\Lambda\| = \|\alpha_0\beta_1\alpha_1 \dots \beta_k\alpha_k\|$ for its norm, and $|\Lambda|_* = k$ for the number of cycles in Λ.

We also treat Λ as the language defined by it, and thus for two configurations $p(\boldsymbol{u})$ and $q(\boldsymbol{v})$ we write $p(\boldsymbol{u}) \xrightarrow{\Lambda}_{\mathbb{D}} q(\boldsymbol{v})$ if and only if there exists $e_1, \dots, e_k \in \mathbb{N}$ such that

$$p(\boldsymbol{u}) \xrightarrow{\alpha_0\beta_1^{e_1}\alpha_1 \dots \beta_k^{e_k}\alpha_k}_{\mathbb{D}} q(\boldsymbol{v}). \tag{6}$$

Positive linear path schemes

A *positive LPS* is a regular expression of the form $\Lambda^+ = \alpha_0\beta_1^+\alpha_1 \dots \beta_k^+\alpha_k$ which is similar to a linear path scheme except that we require each cycle to be used at least once. We write $p(\boldsymbol{u}) \xrightarrow{\Lambda^+}_{\mathbb{D}} q(\boldsymbol{v})$ if and only if there are positive integers $e_1, \dots, e_k \in \mathbb{N}_{>0}$ such that

$$p(\boldsymbol{u}) \xrightarrow{\alpha_0\beta_1^{e_1}\alpha_1 \dots \beta_k^{e_k}\alpha_k}_{\mathbb{D}} q(\boldsymbol{v}). \tag{7}$$

A path π is said to be *admitted by* Λ^+ if $\pi = \alpha_0\beta_1^{e_1}\alpha_1 \dots \beta_k^{e_k}\alpha_k$ for some $e_1, \dots, e_k \in \mathbb{N}_{>0}$. We prefer positive LPSes as they can be easily characterized by linear inequality systems (see Section 4 for details). In fact, positive LPSes can be easily obtained from LPSes:

▶ **Lemma 3.2.** *For every linear path scheme Λ there exists a finite set S of positive linear path schemes compatible to the same VASSes with Λ, such that $\xrightarrow{\Lambda} = \bigcup_{\Lambda^+ \in S} \xrightarrow{\Lambda^+}$ and $|\Lambda^+| \le |\Lambda|$ for every $\Lambda^+ \in S$.*

Proof. Suppose $\Lambda = \alpha_0\beta_1^*\alpha_1 \dots \beta_k^*\alpha_k$. For each cycle component β_i^* in Λ we replace it by either β_i^+ or an empty word nondeterministically. Let S be the set of all resulting positive LPSes. It is obvious that S satisfies the desired properties. ◀

3.2 Main Results

The main results of this section is stated as follows.

▶ **Theorem 3.3.** *Let $G = (Q, T)$ be a geometrically 2-dimensional d-VASS. For every pair of configurations $p(\boldsymbol{u}), q(\boldsymbol{v}) \in Q \times \mathbb{N}^d$ with $p(\boldsymbol{u}) \xrightarrow{*} q(\boldsymbol{v})$ there exists a positive LPS Λ^+ compatible to G such that $p(\boldsymbol{u}) \xrightarrow{\Lambda^+} q(\boldsymbol{v})$ and $|\Lambda^+| \le |G|^{O(1)}$.*

We remark that the big-O term here and elsewhere in the paper hides constant that may depend on the dimension d, but does not depend on $G, \boldsymbol{u}, \boldsymbol{v}$ or anything else.

By Lemma 3.2 we know that positive LPSes can be obtained from LPSes. Thus theorem 3.3 follows easily from the following relaxed theorem, which will be proved in the appendix.

▶ **Theorem 3.4.** *Let $G = (Q, T)$ be a geometrically 2-dimensional d-VASS. For every pair of configurations $p(\boldsymbol{u}), q(\boldsymbol{v}) \in Q \times \mathbb{N}^d$ with $p(\boldsymbol{u}) \xrightarrow{*} q(\boldsymbol{v})$ there exists an LPS Λ compatible to G such that $p(\boldsymbol{u}) \xrightarrow{\Lambda} q(\boldsymbol{v})$ and $|\Lambda| \le |G|^{O(1)}$.*

4 Characteristic Systems for Linear Path Schemes

The property that linear path schemes can be fully characterized by linear inequality systems is exploited in [2] to derive the PSPACE upper bound of the reachability problem in 2-VASSes. Here we recall this linear inequality system and its properties.

We mainly focus on positive linear path schemes. Fix $\Lambda = \alpha_0\beta_1^+\alpha_1 \dots \beta_k^+\alpha_k$ to be a positive LPS from state p to q that is compatible to some d-VASS $G = (Q, T)$, where $k = |\Lambda|_*$ is the number of cycles in Λ.

▶ **Definition 4.1** (cf. [2, Lem. 14]). *The* characteristic system $\mathcal{E}_{\mathrm{LPS}}(\Lambda)$ of the positive LPS Λ *is the system of linear inequalities such that a triple $\boldsymbol{h} = (\boldsymbol{u}, \boldsymbol{e}, \boldsymbol{v}) \in \mathbb{N}^d \times \mathbb{N}^k \times \mathbb{N}^d$ satisfies $\mathcal{E}_{\mathrm{LPS}}(\Lambda)$, written $\boldsymbol{h} \models \mathcal{E}_{\mathrm{LPS}}(\Lambda)$, if and only if the following conditions hold:*

1. *for every $i = 1, \dots, k$, $\boldsymbol{e}(i) \ge 1$;*
2. *for every $i = 0, \dots, k$ and every $j = 1, \dots, |\alpha_i|$,*
$$\boldsymbol{u} + \Delta(\alpha_0\beta_1^{\boldsymbol{e}(1)}\alpha_1 \dots \alpha_{i-1}\beta_i^{\boldsymbol{e}(i)}) + \Delta(\alpha_i[1 \dots j]) \ge \boldsymbol{0}; \tag{8}$$
3. *for every $i = 1, \dots, k$ and every $j = 1, \dots, |\beta_i|$,*
$$\boldsymbol{u} + \Delta(\alpha_0\beta_1^{\boldsymbol{e}(1)}\alpha_1 \dots \beta_{i-1}^{\boldsymbol{e}(i-1)}\alpha_{i-1}) \qquad + \Delta(\beta_i[1 \dots j]) \ge \boldsymbol{0}, \tag{9}$$
$$\boldsymbol{u} + \Delta(\alpha_0\beta_1^{\boldsymbol{e}(1)}\alpha_1 \dots \beta_{i-1}^{\boldsymbol{e}(i-1)}\alpha_{i-1}\beta_i^{\boldsymbol{e}(i)-1}) + \Delta(\beta_i[1 \dots j]) \ge \boldsymbol{0}; \tag{10}$$
4. *and finally, $\boldsymbol{u} + \Delta(\alpha_0\beta_1^{\boldsymbol{e}(1)}\alpha_1 \dots \beta_k^{\boldsymbol{e}(k)}\alpha_k) = \boldsymbol{v}$.*

The readers can easily verify that these constraints are indeed linear in terms of $\boldsymbol{u}, \boldsymbol{e}, \boldsymbol{v}$. The next lemma shows that $\mathcal{E}_{\mathrm{LPS}}(\Lambda)$ indeed captures all runs admitted by Λ.

▶ **Lemma 4.2.** *Let G be a d-VASS and $\Lambda = \alpha_0\beta_1^+\alpha_1 \dots \beta_k^+\alpha_k$ be a positive LPS from state p to q compatible to G. Then for every $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{N}^d$, $p(\boldsymbol{u}) \xrightarrow{\Lambda} q(\boldsymbol{v})$ if and only if there exists $\boldsymbol{e} \in \mathbb{N}^k$ such that $(\boldsymbol{u}, \boldsymbol{e}, \boldsymbol{v}) \models \mathcal{E}_{\mathrm{LPS}}(\Lambda)$. Moreover, for every $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{N}^d$ and every $\boldsymbol{e} \in \mathbb{N}^k$ such that $(\boldsymbol{u}, \boldsymbol{e}, \boldsymbol{v}) \models \mathcal{E}_{\mathrm{LPS}}(\Lambda)$, we have $p(\boldsymbol{u}) \xrightarrow{\alpha_0\beta_1^{\boldsymbol{e}(1)}\alpha_1 \dots \beta_k^{\boldsymbol{e}(k)}\alpha_k} q(\boldsymbol{v})$.*

We also need to introduce the homogeneous version of $\mathcal{E}_{\mathrm{LPS}}(\Lambda)$ for technical reasons.

▶ **Definition 4.3.** *The* homogeneous characteristic system $\mathcal{E}^0_{\mathrm{LPS}}(\Lambda)$ of Λ *is the system of linear inequalities such that a triple $\boldsymbol{h}_0 = (\boldsymbol{u}_0, \boldsymbol{e}_0, \boldsymbol{v}_0) \in \mathbb{N}^d \times \mathbb{N}^k \times \mathbb{N}^d$ satisfies $\mathcal{E}^0_{\mathrm{LPS}}(\Lambda)$, written $\boldsymbol{h}_0 \models \mathcal{E}^0_{\mathrm{LPS}}(\Lambda)$, if and only if the following conditions hold:*

1. *for every $i = 0, \dots, k$, $\boldsymbol{u}_0 + \Delta(\beta_1) \cdot \boldsymbol{e}_0(1) + \dots + \Delta(\beta_i) \cdot \boldsymbol{e}_0(i) \ge \boldsymbol{0}$;*
2. *$\boldsymbol{u}_0 + \Delta(\beta_1) \cdot \boldsymbol{e}_0(1) + \dots + \Delta(\beta_k) \cdot \boldsymbol{e}_0(k) = \boldsymbol{v}_0$.*

5 The Modified KLMST Decomposition Algorithm

In this section we apply our results of Section 3 to improve the notoriously hard KLMST decomposition algorithm for VASS reachability. Our narration will base on the work of Leroux and Schmitz [15]. For readers familiar with [15], the major modifications are listed below:

- The decomposition structure is now a sequence of generalized VASS reachability instances linked by (positive) linear path schemes rather than by single transitions.
- We introduce a new "cleaning" step that replaces all VASS instances which are geometrically 2-dimensional by polynomial-length linear path schemes compatible to them.
- We do not guarantee the exact preservation of action languages at each decomposition step. Instead, we only preserve a subset of action languages. This is a compromise since linear path schemes capture only the reachability relation but not every possible run. Nonetheless, it is enough for the reachability problem.

In this section we focus on the effectiveness and correctness of the modified KLMST decomposition algorithm. Its complexity will be analyzed in Section 6.

5.1 Linear KLM Sequences

The underlying decomposition structure in the KLMST algorithm was known as KLM sequences, named after Mayr[17], Kosaraju[10], and Lambert[11].

▶ **Definition 5.1.** *A* KLM tuple *of dimension d is a tuple $\langle p(\boldsymbol{x})Gq(\boldsymbol{y})\rangle$ where $G = (Q, T)$ is a d-VASS and $p(\boldsymbol{x}), q(\boldsymbol{y}) \in Q \times \mathbb{N}_\omega^d$ are two (generalized) configurations of G. A* KLM sequence *of dimension d is a sequence of KLM tuples interleaved by transitions of the form*

$$\xi = \langle p_0(\boldsymbol{x}_0)G_0q_0(\boldsymbol{y}_0)\rangle\, t_1\, \langle p_1(\boldsymbol{x}_1)G_1q_1(\boldsymbol{y}_1)\rangle\, t_2 \ldots t_k\, \langle p_k(\boldsymbol{x}_k)G_kq_k(\boldsymbol{y}_k)\rangle\,, \tag{11}$$

where each tuple $\langle p_i(\boldsymbol{x}_i)G_iq_i(\boldsymbol{y}_i)\rangle$ is a KLM tuple of dimension d and each t_i is a transition of the form $(q_{i-1}, \boldsymbol{a}_i, p_i)$ from state q_{i-1} to p_i with effect $\boldsymbol{a}_i \in \mathbb{Z}^d$.

In this paper we generalize the definition of KLM sequences to allow (positive) linear path schemes to connect KLM tuples.

▶ **Definition 5.2.** *A* linear KLM sequence *of dimension d is a sequence*

$$\xi = \langle p_0(\boldsymbol{x}_0)G_0q_0(\boldsymbol{y}_0)\rangle\, \Lambda_1\, \langle p_1(\boldsymbol{x}_1)G_1q_1(\boldsymbol{y}_1)\rangle\, \Lambda_2 \ldots \Lambda_k\, \langle p_k(\boldsymbol{x}_k)G_kq_k(\boldsymbol{y}_k)\rangle\,, \tag{12}$$

where each tuple $\langle p_i(\boldsymbol{x}_i)G_iq_i(\boldsymbol{y}_i)\rangle$ is a KLM tuple of dimension d and each Λ_i is a positive linear path scheme from state q_{i-1} to p_i.

One immediately sees that KLM sequences are just special cases of linear KLM sequences. Let ξ be a linear KLM sequence given as (12), we write $\xi_i := \langle p_i(\boldsymbol{x}_i)G_iq_i(\boldsymbol{y}_i)\rangle$ for the ith KLM tuple occurring in ξ.

Action languages

Let ξ be a linear KLM sequence given as (12). We say a path π from state p_0 to q_k is *admitted by* ξ, written $\xi \vdash \pi$, if π can be written as $\pi = \pi_0\rho_1\pi_1 \ldots \rho_k\pi_k$ where π_i is a path from p_i to q_i in G_i for each $i = 0, \ldots, k$, and ρ_i is a path admitted by Λ_i for each $i = 1, \ldots, k$, such that there exist vectors $\boldsymbol{m}_0, \boldsymbol{n}_0, \ldots, \boldsymbol{m}_k, \boldsymbol{n}_k \in \mathbb{N}^d$ such that

$$p_0(\boldsymbol{m}_0) \xrightarrow{\pi_0} q_0(\boldsymbol{n}_0) \xrightarrow{\rho_1} p_1(\boldsymbol{m}_1) \xrightarrow{\pi_1} q_1(\boldsymbol{n}_1) \xrightarrow{\rho_2} \cdots \xrightarrow{\rho_k} p_k(\boldsymbol{m}_k) \xrightarrow{\pi_k} q_k(\boldsymbol{n}_k) \tag{13}$$

and that $\boldsymbol{m}_i \sqsubseteq \boldsymbol{x}_i$, $\boldsymbol{n}_i \sqsubseteq \boldsymbol{y}_i$ for each $i = 0, \ldots, k$.

The *action language* L_ξ of ξ is the language over $\mathbb{Z}^d$ defined by $L_\xi := \{[\![\pi]\!] : \xi \vdash \pi\}$, where we recall that $[\![\cdot]\!]$ is the word morphism mapping each transition to its effect.

We are more interested in the action languages because in some decomposition steps we have to modify the set of transitions, and only the action word of admitted runs can be preserved. Notice that action languages preserve not only the effects of admitted runs, but also their lengths.

Ranks and Sizes

Let t be a transition in a d-VASS G, we define $\mathrm{Cyc}(G/t)$ to be the vector space spanned by the effects of all cycles in G containing t. For the VASS G, let r_i be the number of transitions t in G such that $\dim(\mathrm{Cyc}(G/t)) = i$ for each $i = 0, \ldots, d$. Then the *rank* of G is defined as $\mathrm{rank}(G) = (r_d, \ldots, r_3) \in \mathbb{N}^{d-2}$. We also define the *full rank* of G as $\mathrm{rank}_{\mathrm{full}}(G) = (r_d, \ldots, r_0) \in \mathbb{N}^{d+1}$.

The following lemma was proved in [15], which shows that in a strongly connected VASS G, the space $\mathrm{Cyc}(G/t)$ corresponds to $\mathrm{Cyc}(G)$.

▶ **Lemma 5.3** ([15, Lem. 3.2]). *Let t be a transition of a strongly connected VASS G. Then* $\mathrm{Cyc}(G/t) = \mathrm{Cyc}(G)$.

The following corollary is immediate.

▶ **Corollary 5.4.** *Let G be a strongly connected d-VASS. Then* $\mathrm{rank}(G) = \mathbf{0}$ *if and only if G is geometrically 2-dimensional.*

Let ξ be a linear KLM sequence given as (12). We define the *rank* of ξ as $\mathrm{rank}(\xi) = \sum_{i=0}^{k} \mathrm{rank}(G_i)$, and the *full rank* of ξ as $\mathrm{rank}_{\mathrm{full}}(\xi) = \sum_{i=0}^{k} \mathrm{rank}_{\mathrm{full}}(G_i)$. We remark that the full rank corresponds to the rank defined in [15]. Ranks are ordered lexicographically: let $\boldsymbol{r} = (r_d, \ldots, r_0)$ and $\boldsymbol{r'} = (r'_d, \ldots, r'_0)$, we write $\boldsymbol{r} \leq_{\mathrm{lex}} \boldsymbol{r'}$ if $\boldsymbol{r} = \boldsymbol{r'}$ or the maximal i with $r_i \neq r'_i$ satisfies $r_i < r'_i$.

Recall that for a VASS G we write $|G|$ for its size as defined in (1). For a linear path scheme Λ, its length $|\Lambda|$ and norm $\|\Lambda\|$ are defined in Section 3.1. Let $\zeta = \langle p(\boldsymbol{x})Gq(\boldsymbol{y})\rangle$ be a KLM tuple of dimension d, its *size* is defined to be $|\zeta| = |G| + d \cdot (\|\boldsymbol{x}\| + \|\boldsymbol{y}\| + 1)$. Let ξ be a linear KLM sequence given as (12), we define its *size* as

$$|\xi| = \sum_{i=0}^{k} |\xi_i| + \sum_{i=1}^{k} d \cdot |\Lambda_i| \cdot (\|\Lambda_i\| + 1). \tag{14}$$

Note that the sizes defined in this paper reflect the sizes of unary encoding, thus have an exponential expansion in their binary encoding.

5.2 Characteristic Systems for Linear KLM Sequences

We define in this section the characteristic systems of linear KLM sequences, which are systems of linear inequalities that serve as an under-specification of admitted runs. Let $G = (Q, T)$ be a VASS, we first recall the *Kirchhoff system* $K_{G,p,q}$ of G with respect to states $p, q \in Q$, which is a system of linear equations such that a function $\phi \in \mathbb{N}^T$ is a model of $K_{G,p,q}$, written $\phi \models K_{G,p,q}$, if and only if

$$\mathbf{1}_q - \mathbf{1}_p = \sum_{t=(r,\boldsymbol{a},s)\in T} \phi(t) \cdot (\mathbf{1}_s - \mathbf{1}_r), \tag{15}$$

where $\mathbf{1}_p \in \{0,1\}^Q$ is the indicator function defined by $\mathbf{1}_p(q) = 1$ if $q = p$ and $\mathbf{1}_p(q) = 0$ otherwise. We also need the homogeneous version of $K_{G,p,q}$, denoted by $K^0_{G,p,q}$, where a function $\phi \in \mathbb{N}^T$ is a model of it, written $\phi \models K^0_{G,p,q}$, if and only if

$$\mathbf{0} = \sum_{t=(r,\boldsymbol{a},s)\in T} \phi(t) \cdot (\mathbf{1}_s - \mathbf{1}_r). \tag{16}$$

▶ **Definition 5.5.** *Let ξ be a linear KLM sequence given by*

$$\xi = \langle p_0(\boldsymbol{x}_0)G_0q_0(\boldsymbol{y}_0)\rangle\, \Lambda_1\, \langle p_1(\boldsymbol{x}_1)G_1q_1(\boldsymbol{y}_1)\rangle\, \Lambda_2 \ldots \Lambda_k\, \langle p_k(\boldsymbol{x}_k)G_kq_k(\boldsymbol{y}_k)\rangle \tag{17}$$

The characteristic system $\mathcal{E}(\xi)$ *is a set of linear (in)equalities such that a sequence*

$$\boldsymbol{h} = (\boldsymbol{m}_0, \phi_0, \boldsymbol{n}_0), \boldsymbol{e}_1, (\boldsymbol{m}_1, \phi_1, \boldsymbol{n}_1), \boldsymbol{e}_2, \ldots, \boldsymbol{e}_k, (\boldsymbol{m}_k, \phi_k, \boldsymbol{n}_k), \tag{18}$$

where each $(\boldsymbol{m}_i, \phi_i, \boldsymbol{n}_i) \in \mathbb{N}^d \times \mathbb{N}^{T_i} \times \mathbb{N}^d$ and each $\boldsymbol{e}_i \in \mathbb{N}^{|\Lambda_i|_*}$, is a model of $\mathcal{E}(\xi)$, written $\boldsymbol{h} \models \mathcal{E}(\xi)$, if and only if

1. $\boldsymbol{m}_i \sqsubseteq \boldsymbol{x}_i$, $\phi_i \models K_{G,p,q}$, $\boldsymbol{n}_i \sqsubseteq \boldsymbol{y}_i$ *and* $\boldsymbol{n}_i = \boldsymbol{m}_i + \Delta(\phi_i)$ *for every* $i = 0, \ldots, k$;
2. $(\boldsymbol{n}_{i-1}, \boldsymbol{e}_i, \boldsymbol{m}_i) \models \mathcal{E}_{\mathrm{LPS}}(\Lambda_i)$ *for every* $i = 1, \ldots, k$.

Similarly, the homogeneous characteristic system $\mathcal{E}^0(\xi)$ *is a set of linear (in)equalities such that a sequence $\boldsymbol{h}$ of the form (18) is a model of $\mathcal{E}^0(\xi)$, written $\boldsymbol{h} \models \mathcal{E}^0(\xi)$, if and only if*

1. $\boldsymbol{m}_i(j) = 0$ *whenever* $\boldsymbol{x}_i(j) \neq \omega$, $\phi_i \models K^0_{G,p,q}$, $\boldsymbol{n}_i(j) = 0$ *whenever* $\boldsymbol{y}_i(j) \neq \omega$, *and* $\boldsymbol{n}_i = \boldsymbol{m}_i + \Delta(\phi_i)$ *for every* $i = 0, \ldots, k$;
2. $(\boldsymbol{n}_{i-1}, \boldsymbol{e}_i, \boldsymbol{m}_i) \models \mathcal{E}^0_{\mathrm{LPS}}(\Lambda_i)$ *for every* $i = 1, \ldots, k$.

The sequence ξ is said to be satisfiable *if $\mathcal{E}(\xi)$ has a model, otherwise it's unsatisfiable.*

Let $\boldsymbol{h}$ be a model of $\mathcal{E}(\xi)$ (or $\mathcal{E}^0(\xi)$), we shall write $\boldsymbol{m}_i^{\boldsymbol{h}}, \phi_i^{\boldsymbol{h}}, \boldsymbol{n}_i^{\boldsymbol{h}}, \boldsymbol{e}_i^{\boldsymbol{h}}$ for the values of $\boldsymbol{m}_i, \phi_i, \boldsymbol{n}_i, \boldsymbol{e}_i$ assigned by $\boldsymbol{h}$, respectively. Recall that unsatisfiable linear KLM sequences have empty action languages.

▶ **Lemma 5.6** (cf. [15, Lem. 3.5]). *For any unsatisfiable linear KLM sequence ξ, $L_\xi = \emptyset$.*

5.2.1 Bounds on Bounded Values in $\mathcal{E}(\xi)$

We state here a lemma similar to [15, Lem. 3.7], which upper bounds the bounded values in the characteristic system $\mathcal{E}(\xi)$. Its proof can be found in the appendix, which is a straightforward application of tools in [3] and [18].

▶ **Lemma 5.7.** *Assume that $\xi = \langle p_0(\boldsymbol{x}_0)G_0q_0(\boldsymbol{y}_0)\rangle\, \Lambda_1 \cdots \Lambda_k\, \langle p_k(\boldsymbol{x}_k)G_kq_k(\boldsymbol{y}_k)\rangle$ is satisfiable. Then for every $0 \leq i \leq k$ we have:*

- *For every $1 \leq j \leq d$, the set of values $\boldsymbol{m}_i^{\boldsymbol{h}}(j)$ where $\boldsymbol{h} \models \mathcal{E}(\xi)$ is unbounded if, and only if, there exists a model $\boldsymbol{h}_0$ of $\mathcal{E}^0(\xi)$ such that $\boldsymbol{m}_i^{\boldsymbol{h}_0}(j) > 0$.*
- *For every $t \in T_i$, the set of values $\phi_i^{\boldsymbol{h}}(t)$ where $\boldsymbol{h} \models \mathcal{E}(\xi)$ is unbounded if, and only if, there exists a model $\boldsymbol{h}_0$ of $\mathcal{E}^0(\xi)$ such that $\phi_i^{\boldsymbol{h}}(t) > 0$.*
- *For every $1 \leq j \leq d$, the set of values $\boldsymbol{n}_i^{\boldsymbol{h}}(j)$ where $\boldsymbol{h} \models \mathcal{E}(\xi)$ is unbounded if, and only if, there exists a model $\boldsymbol{h}_0$ of $\mathcal{E}^0(\xi)$ such that $\boldsymbol{n}_i^{\boldsymbol{h}_0}(j) > 0$.*

Moreover, every bounded value of $\mathcal{E}(\xi)$ is bounded by $(10|\xi|)^{12|\xi|}$.

5.3 Cleaning of Linear KLM Sequences

In this section we define three conditions that require a linear KLM sequence to be *strongly connected*, *pure*, and *saturated*. Together with the satisfiability condition, they make up the so-called "clean" condition of linear KLM sequences. Note that the purity condition is new compared to [15], which requires every geometrically 2-dimensional VASSes occur in a linear KLM sequence to be replaced by linear path schemes.

Strongly Connected Sequences

A linear KLM sequence $\xi = \langle p_0(\boldsymbol{x}_0)G_0q_0(\boldsymbol{y}_0)\rangle\, \Lambda_1 \cdots \Lambda_k\, \langle p_k(\boldsymbol{x}_k)G_kq_k(\boldsymbol{y}_k)\rangle$ is *strongly connected* if all the VASSes $G_0, \ldots, G_k$ are strongly connected (as they are understood as directed graphs). One can easily obtain strongly connected sequences by expanding the strongly connected components of each VASS:

▶ **Lemma 5.8** ([15, Lem. 4.2]). *For any linear KLM sequence ξ which is not strongly connected, one can compute in time* $\mathsf{exp}(|\xi|)$ *a finite set Ξ of strongly connected linear KLM sequences such that $L_\xi = \bigcup_{\xi' \in \Xi} L_{\xi'}$ and that* $\mathrm{rank}(\xi') \leq_{lex} \mathrm{rank}(\xi)$ *and* $|\xi'| \leq (2d+1)|\xi|$ *for every $\xi' \in \Xi$.*

Pure Sequences

A KLM tuple $\langle p(\boldsymbol{x})Gq(\boldsymbol{y})\rangle$ is called *trivial* if $p(\boldsymbol{x}) = q(\boldsymbol{y})$ and G contains no transition and only a single state p. In this case we simply write $\langle p(\boldsymbol{x})\rangle$ for this tuple. Note that the action language of a trivial tuple contains exactly the empty word.

A linear KLM sequence $\xi = \langle p_0(\boldsymbol{x}_0)G_0q_0(\boldsymbol{y}_0)\rangle\, \Lambda_1 \cdots \Lambda_k\, \langle p_k(\boldsymbol{x}_k)G_kq_k(\boldsymbol{y}_k)\rangle$ is said to be *pure* if ξ is strongly connected and for every $i = 0, \ldots, k$, $\mathrm{rank}(G_i) = \mathbf{0}$ implies that the tuple $\langle p_i(\boldsymbol{x}_i)G_iq_i(\boldsymbol{y}_i)\rangle$ is trivial. By Corollary 5.4, a rank-$\mathbf{0}$ strongly connected VASS is geometrically 2-dimensional, and thus can be replaced by linear path schemes in case it is not trivial.

▶ **Lemma 5.9.** *Let ξ be a strongly connected linear KLM sequence. Whether ξ is pure is in* PSPACE. *If ξ is not pure, one can compute in space* $\mathsf{poly}(|\xi|)$ *a finite set Ξ of pure linear KLM sequences such that $\bigcup_{\xi' \in \Xi} L_{\xi'} \subseteq L_\xi$ and $\bigcup_{\xi' \in \Xi} L_{\xi'} \neq \emptyset$ whenever $L_\xi \neq \emptyset$, and such that* $\mathrm{rank}(\xi') = \mathrm{rank}(\xi)$ *and* $|\xi'| \leq |\xi|^{O(1)}$ *for all $\xi' \in \Xi$.*

Saturated Sequences

Let $\xi = \langle p_0(\boldsymbol{x}_0)G_0q_0(\boldsymbol{y}_0)\rangle\, \Lambda_1 \cdots \Lambda_k\, \langle p_k(\boldsymbol{x}_k)G_kq_k(\boldsymbol{y}_k)\rangle$ be a linear KLM sequence. We say ξ is *saturated* if for every $0 \leq i \leq k$ and every $j \in [d]$, we have

- $\boldsymbol{x}_i(j) = \omega$ implies the set of values $\boldsymbol{m}_i^{\boldsymbol{h}}(j)$ where $\boldsymbol{h} \models \mathcal{E}(\xi)$ is unbounded; and
- $\boldsymbol{y}_i(j) = \omega$ implies the set of values $\boldsymbol{n}_i^{\boldsymbol{h}}(j)$ where $\boldsymbol{h} \models \mathcal{E}(\xi)$ is unbounded.

▶ **Lemma 5.10** ([15, Lem. 4.4]). *From any pure linear KLM sequence ξ, one can compute in time* $\mathsf{exp}(|\xi|^{O(|\xi|)})$ *a finite set Ξ of saturated pure linear KLM sequences such that $L_\xi = \bigcup_{\xi' \in \Xi} L_{\xi'}$, and such that* $\mathrm{rank}(\xi') = \mathrm{rank}(\xi)$ *and* $|\xi'| \leq |\xi|^{O(|\xi|)}$ *for every $\xi' \in \Xi$.*

Proof. By Lemma 5.7, if a variable $\boldsymbol{m}_i(j)$ or $\boldsymbol{n}_i(j)$ is bounded in $\mathcal{E}(\xi)$, we can replace the corresponding ω component in ξ by all possible values bounded by $(10|\xi|)^{12|\xi|} \leq |\xi|^{O(|\xi|)}$. ◀

The Cleaning Lemma

A linear KLM sequence ξ is called *clean* if it is satisfiable, strongly connected, pure and saturated. The lemmas 5.8 through 5.10 show how to make a linear KLM sequence clean.

▶ **Lemma 5.11.** *From any linear KLM sequence ξ, one can compute in time* $\mathsf{exp}(g(|\xi|))$ *a finite set* $\mathrm{clean}(\xi)$ *of clean linear KLM sequences such that $\bigcup_{\xi' \in \mathrm{clean}(\xi)} L_{\xi'} \subseteq L_\xi$ and $\bigcup_{\xi' \in \mathrm{clean}(\xi)} L_{\xi'} \neq \emptyset$ whenever $L_\xi \neq \emptyset$. Moreover, for every $\xi' \in \mathrm{clean}(\xi)$ we have* $\mathrm{rank}(\xi') \leq_{lex} \mathrm{rank}(\xi)$ *and* $|\xi'| \leq g(|\xi|)$ *where* $g(x) = x^{x^{O(1)}}$.

5.4 Decomposition of Linear KLM Sequences

In this section we recall three conditions that require a linear KLM sequence to be *unbounded*, *rigid*, and *pumpable*. If any one of them is violated, a decomposition into a set of linear KLM sequences with strictly smaller ranks can be performed. Essentially there is nothing new in this section compared to [15]. The decomposition operations in [15] can be directly applied here, since they operate on a single KLM tuple and produce KLM sequences that are just special cases of linear KLM sequences. The proofs in [15] can also be adapted easily, and we will omit the details here. Especially, the next lemma shows that the arguments of strict decrease in ranks are still valid even though we discard the lower three components of ranks.

▶ **Lemma 5.12.** *Let ξ' be a pure linear KLM sequence. For any linear KLM sequence ξ' with $\mathrm{rank}_{full}(\xi') <_{lex} \mathrm{rank}_{full}(\xi)$, we have $\mathrm{rank}(\xi') <_{lex} \mathrm{rank}(\xi)$.*

Unbounded Sequences

Let $\xi = \langle p_0(\boldsymbol{x}_0)G_0q_0(\boldsymbol{y}_0)\rangle\, \Lambda_1 \cdots \Lambda_k\, \langle p_k(\boldsymbol{x}_k)G_kq_k(\boldsymbol{y}_k)\rangle$ be a linear KLM sequence. We say ξ is *unbounded* if for all $i = 0, \ldots, k$ and every transition $t \in T_i$ where T_i is the set of transitions of G_i, the set of values $\phi_i^{\boldsymbol{h}}(t)$ where $\boldsymbol{h} \models \mathcal{E}(\xi)$ is unbounded. Bounded transitions can be expanded exhaustively according to the bounds given by Lemma 5.7.

▶ **Lemma 5.13** ([15, Lem. 4.6]). *Whether a linear KLM sequence ξ is unbounded is decidable in* NP*. Moreover, if ξ is pure and bounded, one can compute in time $\mathsf{exp}(|\xi|^{O(|\xi|)})$ a finite set Ξ of linear KLM sequences such that $L_\xi = \bigcup_{\xi' \in \Xi} L_{\xi'}$ and such that $\mathrm{rank}(\xi') <_{lex} \mathrm{rank}(\xi)$ and $|\xi'| < |\xi|^{O(|\xi|)}$ for every $\xi' \in \Xi$.*

Rigid Sequences

A coordinate $j \in [d]$ is said to be *fixed* by a VASS $G = (Q, T)$ if there exists a function $f_j : Q \to \mathbb{N}$ such that $f_j(q) = f_j(p) + \boldsymbol{a}(j)$ for every transition $(p, \boldsymbol{a}, q) \in T$. We also say that f_j *fixes* G at coordinate j in this case.

A KLM tuple $\langle p(\boldsymbol{x})Gq(\boldsymbol{y})\rangle$ is said to be *rigid* if for every coordinate j fixed by $G = (Q, T)$, there exists a function $g_j : Q \to \mathbb{N}$ that fixes G at coordinate j and such that $g_j(p) \sqsubseteq \boldsymbol{x}(j)$ and $g_j(q) \sqsubseteq \boldsymbol{y}(j)$.

A linear KLM sequence $\xi = \langle p_0(\boldsymbol{x}_0)G_0q_0(\boldsymbol{y}_0)\rangle\, \Lambda_1 \cdots \Lambda_k\, \langle p_k(\boldsymbol{x}_k)G_kq_k(\boldsymbol{y}_k)\rangle$ is said to be *rigid* if every tuple $\langle p_i(\boldsymbol{x}_i)G_iq_i(\boldsymbol{y}_i)\rangle$ in ξ is rigid.

▶ **Lemma 5.14** ([15, Lem. 4.9]). *From any pure linear KLM sequence ξ one can decide in polynomial time whether ξ is not rigid. Moreover, in that case one can compute in polynomial time a linear KLM sequence ξ' such that $L_\xi = L_{\xi'}$, $\mathrm{rank}(\xi') <_{lex} \mathrm{rank}(\xi)$, and $|\xi'| \le |\xi|$.*

Pumpable Sequences

Given a KLM tuple $\langle p(\boldsymbol{x})Gq(\boldsymbol{y})\rangle$, recall the *forward and backward acceleration vectors* $\textsc{Facc}_{G,p}(\boldsymbol{x}), \textsc{Bacc}_{G,q}(\boldsymbol{y}) \in \mathbb{N}_\omega^d$ defined by

$$\textsc{Facc}_{G,p}(\boldsymbol{x})(j) = \begin{cases} \omega & \text{if } p(\boldsymbol{x}) \xrightarrow{*} p(\boldsymbol{x}') \text{ for some } \boldsymbol{x}' \text{ with } \boldsymbol{x}' \ge \boldsymbol{x}, \boldsymbol{x}'(j) > \boldsymbol{x}(j) \\ \boldsymbol{x}(j) & \text{otherwise} \end{cases} \tag{19}$$

$$\textsc{Bacc}_{G,q}(\boldsymbol{y})(j) = \begin{cases} \omega & \text{if } q(\boldsymbol{y}') \xrightarrow{*} q(\boldsymbol{y}) \text{ for some } \boldsymbol{y}' \text{ with } \boldsymbol{y}' \ge \boldsymbol{y}, \boldsymbol{y}'(j) > \boldsymbol{y}(j) \\ \boldsymbol{y}(j) & \text{otherwise} \end{cases} \tag{20}$$

A tuple $\langle p(\boldsymbol{x})Gq(\boldsymbol{y})\rangle$ is said to be *pumpable* if $\text{FACC}_{G,p}(\boldsymbol{x})(j) = \text{BACC}_{G,q}(\boldsymbol{y})(j) = \omega$ for every coordinate j *not* fixed by G.

A linear KLM sequence given by $\xi = \langle p_0(\boldsymbol{x}_0)G_0q_0(\boldsymbol{y}_0)\rangle\, \Lambda_1 \cdots \Lambda_k\, \langle p_k(\boldsymbol{x}_k)G_kq_k(\boldsymbol{y}_k)\rangle$ is said to be *pumpable* if every tuple $\langle p_i(\boldsymbol{x}_i)G_iq_i(\boldsymbol{y}_i)\rangle$ in ξ is pumpable.

▶ **Lemma 5.15** ([15, Lem. 4.15])**.** *Whether a linear KLM sequence ξ is pumpable is decidable in* EXPSPACE*. Moreover, if ξ is pure and unpumpable, one can compute in time $\mathsf{exp}(|\xi|^{O(1)})$ a finite set Ξ of linear KLM sequences such that $L_\xi = \bigcup_{\xi' \in \Xi} L_{\xi'}$ and such that $\text{rank}(\xi') <_{lex} \text{rank}(\xi)$ and $|\xi'| < |\xi|^{O(1)}$ for every $\xi' \in \Xi$.*

Note that the $O(1)$ term here hides a constant depending on d, which essentially arises from a result on the coverability problem by Rackoff [19]. The $O(1)$ term also captures the difference between the sizes of linear KLM sequences defined here and that in [15].

The Decomposition Lemma

A linear KLM sequence is *normal* if it is clean, unbounded, rigid, and pumpable. The lemmas 5.13 through 5.15 show that when a clean linear KLM sequence is not normal, we are able to decompose it into a finite set of linear KLM sequences of strictly smaller ranks.

▶ **Lemma 5.16.** *From any clean linear KLM sequences ξ that is not normal, one can compute in time $\mathsf{exp}(h(\xi))$ a finite set $\text{dec}(\xi)$ of clean linear KLM sequence such that $\bigcup_{\xi' \in \text{dec}(\xi)} L_{\xi'} \subseteq L_\xi$ and $\bigcup_{\xi' \in \text{dec}(\xi)} L_{\xi'} \neq \emptyset$ whenever $L_\xi \neq \emptyset$. Moreover, for every $\xi' \in \text{dec}(\xi)$ we have $\text{rank}(\xi') <_{lex} \text{rank}(\xi)$ and $|\xi'| \le h(\xi)$ where $h(x) = x^{x^{x^{O(1)}}}$.*

5.5 Normal Sequences

The following lemma shows that a normal linear KLM sequence is guaranteed to have non-empty action language, thus one can terminate the decomposition process once a normal sequence is produced.

▶ **Lemma 5.17.** *Let ξ be a normal linear KLM sequence, then there is a word $\sigma \in L_\xi$ whose length is bounded by $|\sigma| \le \ell(|\xi|)$ where $\ell(x) \le x^{O(x)}$.*

5.6 Putting All Together: The Modified KLMST Algorithm

Here we describe the modified KLMST decomposition algorithm for VASS reachability problem. Suppose we are given a d-VASS $G = (Q, T)$ and two configurations $p(\boldsymbol{m}), q(\boldsymbol{n}) \in Q \times \mathbb{N}^d$. To decide whether $p(\boldsymbol{m}) \xrightarrow{*} q(\boldsymbol{n})$ holds in G, it is enough to decide whether L_ξ is non-empty where $\xi = \langle p(\boldsymbol{m})Gq(\boldsymbol{n})\rangle$. To start with, we use Lemma 5.11 to clean the sequence ξ, and then choose $\xi^0 \in \text{clean}(\xi)$ non-deterministically. If ξ^0 is normal then we are done by Lemma 5.17. Otherwise, we decompose ξ^0 using Lemma 5.16 and choose $\xi^1 \in \text{dec}(\xi^0)$ non-deterministically. The procedure continues to produce a series of linear KLM sequences $\xi^0, \xi^1, \xi^2, \ldots$ where $\xi^{i+1} \in \text{dec}(\xi^i)$, until either we finally obtain a normal sequence ξ^n, or at some point we have to abort because the decomposition of a linear KLM sequence is the empty set. The procedure terminates because $\text{rank}(\xi^0) >_{\text{lex}} \text{rank}(\xi^1) >_{\text{lex}} \text{rank}(\xi^2) >_{\text{lex}} \cdots$ form a decreasing chain of the well-order $(\mathbb{N}^{d-2}, <_{\text{lex}})$, which must be finite. If $L_\xi = \emptyset$ then we cannot get a normal sequence since the action languages $L_\xi \supseteq L_{\xi^0} \supseteq L_{\xi^1} \supseteq \cdots$ are all empty. On the other hand, if $L_\xi \neq \emptyset$ then there are non-deterministic choices that always choose the linear KLM sequences with non-empty action languages, which finally lead to a normal sequence. This shows the correctness of the algorithm.

6 Complexity Upper Bound

The termination of the modified KLMST decomposition algorithm is guaranteed by a ranking function that decreases along a well-ordering. In order to analyze the length of this decreasing chain, we recall the so-called “length function theorems” by Schmitz [20] in Section 6.1. After that, we can locate the complexity upper bound of the algorithm in the fast-growing complexity hierarchy [21] which we recall in Section 6.2. Readers familiar with [15] may realize that the complexity upper bound for d-VASS can be improved to F_{d+1}, i.e. the $(d{+}1)$-th level in the fast-growing hierarchy, with our ranking function. In fact, by a careful analysis on a property of fast-growing functions, we further improve this bound to F_d.

In this section we assume some familiarity with ordinal numbers (see, e.g. [9]). We write ω here for the first infinite ordinal, not to be confused with the infinite element in previous sections.

6.1 Length of Sequences of Decreasing Ranks

Let ξ be a linear KLM sequence of dimension d with $\operatorname{rank}(\xi) = (r_d, \dots, r_3)$, we define the *ordinal rank* α_ξ of ξ as the ordinal number given by

$$\alpha_\xi := \omega^{d-3} \cdot r_d + \omega^{d-4} \cdot r_{d-1} + \cdots + \omega^0 \cdot r_3. \tag{21}$$

Note that $\operatorname{rank}(\xi) <_{\text{lex}} \operatorname{rank}(\xi')$ if and only if $\alpha_\xi < \alpha_{\xi'}$. With this reformulation, we now focus on the decreasing chain of ordinal ranks.

Let $\alpha < \omega^\omega$ be an ordinal given in Cantor Normal Form as $\alpha = \omega^n \cdot c_n + \cdots + \omega^0 \cdot c_0$ where $n, c_0, \dots, c_n \in \mathbb{N}$, we define the *size* of α as $N\alpha := \max\{n, \max_{0 \le i \le n} c_i\}$. For the linear KLM sequence ξ with $\operatorname{rank}(\xi) = (r_d, \dots, r_3)$, we have $N\alpha_\xi = \max\{d-3, \max_{3 \le i \le d} r_i\} \le |\xi|$.

Given a number $n_0 \in \mathbb{N}$ and a function $h : \mathbb{N} \to \mathbb{N}$ that is monotone inflationary (that is, $x \le h(x)$ and $h(x) \le h(y)$ whenever $x \le y$), we say a sequence of ordinals $\alpha_0, \alpha_1, \dots$ is (n_0, h)-*controlled* if $N\alpha_i \le h^i(n_0)$ for all $i \in \mathbb{N}$, where $h^i(n_0)$ is the ith iteration of h on n_0.

Let $\xi^0, \xi^1, \dots$ be the linear KLM sequences produced in the modified KLMST algorithm, by Lemma 5.16 we know that the sequence of ordinal ranks

$$\alpha_{\xi^0} > \alpha_{\xi^1} > \cdots \tag{22}$$

is $(|\xi^0|, h)$-controlled where h is defined in Lemma 5.16. Recall that $\xi^0 \in \operatorname{clean}(\xi)$ where $\xi = \langle p(\boldsymbol{m})Gq(\boldsymbol{n})\rangle$ corresponds to the input reachability instance. Then $|\xi^0| \le g(|\xi|)$ where g is defined in Lemma 5.11, and (22) is indeed $(g(n), h)$-controlled where $n := |\langle p(\boldsymbol{m})Gq(\boldsymbol{n})\rangle|$.

Length function theorem

The length of the controlled sequence of ordinals (22) can be bounded in terms of the hierarchies of fast-growing functions of Hardy and Cichoń [4]. First recall that given a limit ordinal $\lambda \le \omega^\omega$, the standard *fundamental sequence* of λ is a sequence $(\lambda(x))_{x<\omega}$ defined inductively by

$$\omega^\omega(x) := \omega^{x+1}, \qquad (\beta + \omega^{k+1})(x) := \beta + \omega^k \cdot (x+1) \tag{23}$$

where $\beta + \omega^{k+1}$ is in Cantor Normal Form. Now given a function $h : \mathbb{N} \to \mathbb{N}$ that is monotone inflationary, we define the *Hardy hierarchy* $(h^\alpha)_{\alpha \le \omega^\omega}$ and the *Cichoń hierarchy* $(h_\alpha)_{\alpha \le \omega^\omega}$ as two families of functions $h^\alpha, h_\alpha : \mathbb{N} \to \mathbb{N}$ indexed by ordinals $\alpha \le \omega^\omega$ given inductively by

$$h^0(x) := x, \qquad h^{\alpha+1}(x) := h^\alpha(h(x)), \qquad h^\lambda(x) := h^{\lambda(x)}(x), \tag{24}$$

$$h_0(x) := 0, \qquad h_{\alpha+1}(x) := 1 + h_\alpha(h(x)), \qquad h_\lambda(x) := h_{\lambda(x)}(x). \tag{25}$$

Observe that Cichoń hierarchy counts the number of iterations of h in Hardy hierarchy, that is, $h^{h_\alpha(x)}(x) = h^\alpha(x)$. Also note that as h is monotone inflationary, by induction on α we have $h_\alpha(x) \le h^\alpha(x)$. Now we give the length function theorem as follows.

▶ **Theorem 6.1** (Length function theorem, [20, Thm. 3.3]). *Let $n_0 \ge d-2$, then the maximal length of (n_0, h)-controlled decreasing sequences of ordinals in ω^{d-2} is $h_{\omega^{d-2}}(n_0)$.*

Small witness property

By Theorem 6.1 we can bound the length of (22), which then yields a bound on the minimal length of runs witnessing reachability.

▶ **Lemma 6.2** (Small witnesses). *Let $G = (Q, T)$ be a d-VASS where $d \ge 3$, let $p(\boldsymbol{m}), q(\boldsymbol{n}) \in Q \times \mathbb{N}^d$ be two configurations, and let $n := |\langle p(\boldsymbol{m}) G q(\boldsymbol{n})\rangle|$. If $p(\boldsymbol{m}) \xrightarrow{*} q(\boldsymbol{n})$ holds in G, then there is a path σ such that $p(\boldsymbol{m}) \xrightarrow{\sigma} q(\boldsymbol{n})$ and $|\sigma| \le \ell(h^{\omega^{d-2}}(g(n)))$, where g, h, ℓ are defined in lemmas 5.11, 5.16, and 5.17.*

Proof. Suppose $p(\boldsymbol{m}) \xrightarrow{*} q(\boldsymbol{n})$, then there is a sequence of linear KLM sequence $\xi^0, \xi^1, \ldots, \xi^L$ produced in the modified KLMST algorithm, such that ξ^L is normal. We have discussed that the sequence of their ordinal ranks $\alpha_{\xi^0} > \alpha_{\xi^1} > \cdots > \alpha_{\xi^L}$ is $(g(n), h)$-controlled, so by Theorem 6.1 we have $L \le h_{\omega^{d-2}}(g(n))$. From Lemma 5.16 and the fact that $h^{h_\alpha(x)} = h^\alpha(x)$, the size of ξ^L is bounded by

$$|\xi^L| \le h^L(|\xi^0|) \le h^{h_{\omega^{d-2}}(g(n))}(g(n)) = h^{\omega^{d-2}}(g(n)). \tag{26}$$

Now Lemma 5.17 bounds the length of the minimal witnesses by $\ell(h^{\omega^{d-2}}(g(n)))$. ◀

6.2 Fast-Growing Complexity Hierarchy

We recall the fast-growing hierarchy formally introduced by Schmitz [21] that captures the complexity class high above elementary. Define $H(x) := x + 1$, we shall use the Hardy hierarchy $(H^\alpha)_{\alpha<\omega^\omega}$, where for example $H^{\omega^2}(x) = 2^{x+1}(x+1)$ and $H^{\omega^3}(x)$ grows faster than the tower function. First we define the family $\mathscr{F}_\alpha := \bigcup_{\beta<\omega^\alpha} \mathsf{FDTIME}(H^\beta(n))$ which contains functions computable in deterministic time $O(H^\beta(n))$. Observe that, for example, $\mathscr{F}_3$ contains exactly the Kalmar elementary functions. Now we define

$$\mathsf{F}_\alpha := \bigcup_{p\in\mathscr{F}_\alpha} \mathsf{DTIME}(H^{\omega^\alpha}(p(n))) \tag{27}$$

which is the class of decision problems solvable in deterministic time $O(H^{\omega^\alpha}(p(n)))$. Note that non-deterministic time Turing machines can be made deterministic with an exponential overhead in $\mathscr{F}_3$, thus for $\alpha \ge 3$, we have equivalently that $\mathsf{F}_\alpha = \bigcup_{p\in\mathscr{F}_\alpha} \mathsf{NDTIME}(H^{\omega^\alpha}(p(n)))$. Observe that F_α is closed under reductions in $\mathscr{F}_\alpha$.

6.2.1 Relativized Fast-Growing Functions

In order to express the complexity of the modified KLMST algorithm in terms of the hierarchy $(\mathsf{F}_\alpha)_{\alpha<\omega}$, one needs to locate the function $h^{\omega^{d-2}}$ in the Hardy hierarchy $(H^\alpha)_{\alpha<\omega^\omega}$ where $h \in \mathscr{F}_3$ is the elementary function from Lemma 5.16. Previously we can upper bound $h^{\omega^{d-2}}$ by $H^{\omega^{d+1}}$ with the help of [21, Lem. 4.2]. Here we show a slightly better result, from which we can bound $h^{\omega^{d-2}}(x)$ by $H^{\omega^d}(O(x))$.

▶ **Lemma 6.3** (cf. [21, Lem. A.5]). *Let $h : \mathbb{N} \to \mathbb{N}$ be a monotone inflationary function, let $a, b, c \ge 1$ and $x_0 \ge 0$ be natural numbers. If for all $x \ge x_0$, $h(x) \le H^{\omega^b \cdot c}(x)$, then $h^{\omega^a}(x) \le H^{\omega^{b+a}}((c+1)x)$ for all $x \ge \max\{2c, x_0\}$.*

6.3 Upper Bounds for VASS Reachability

Now we analyze the time complexity of the modified KLMST algorithm. Given as input the d-VASS $G = (Q, T)$ and two configurations $p(\boldsymbol{m}), q(\boldsymbol{n})$, let $\xi := \langle p(\boldsymbol{m})Gq(\boldsymbol{n})\rangle$ and $n := |\xi|$. The initial sequence $\xi^0 \in \mathrm{clean}(\xi)$ can be computed in (non-deterministic) time elementary in n by Lemma 5.11. Then the algorithm produces $\xi^0, \xi^1, \ldots, \xi^L$ with $L \leq h_{\omega^{d-2}}(g(n))$, where g, h are defined in lemmas 5.11, 5.16. Note that in each step, the sequence $\xi^{i+1} \in \mathrm{dec}(\xi^i)$ can be computed in time elementary in $|\xi^i|$ by Lemma 5.16, and the sizes $|\xi^i|$ are all bounded by $h^{\omega^{d-2}}(g(n))$ as we have discussed above in the proof of Lemma 6.2. To sum up, the entire algorithm finishes in non-deterministic time elementary in $h^{\omega^{d-2}}(g(n))$.

▶ **Lemma 6.4.** *On input a d-VASS $G = (Q, T)$ where $d \geq 3$ and $p(\boldsymbol{m}), q(\boldsymbol{n}) \in Q \times \mathbb{N}^d$, the modified KLMST algorithm finishes in non-deterministic time $e(h^{\omega^{d-2}}(g(n)))$ where $n = |\langle p(\boldsymbol{m})Gq(\boldsymbol{n})\rangle|$, g, h are defined in lemmas 5.11, 5.16, and $e \in \mathscr{F}_3$ is some fixed function.*

Since h is an elementary function, there is a number $c \in \mathbb{N}$ such that h is eventually dominated by $H^{\omega^2 \cdot c}$. By Lemma 6.3 we can upper bound $h^{\omega^{d-2}}(x)$ by $H^{\omega^d}((c+1)x)$. Observe that the inner part $g(n)$ is elementary in the binary encoding size of the input $G, p(\boldsymbol{m}), q(\boldsymbol{n})$, thus can be captured by a function $p \in \mathscr{F}_3$. Finally, [21, Lem. 4.6] allows us to move the outermost function e to the innermost position. Hence we have the following upper bound.

▶ **Theorem 6.5.** *Reachability in d-dimensional VASS is in F_d for all $d \geq 3$.*

Also, by Lemma 6.2 there is a simple combinatorial algorithm for d-VASS reachability. We fist compute the bound $B := \ell(h^{\omega^{d-2}}(g(n)))$, which can be done in time elementary in B by [21, Thm. 5.1]. Then we can decide reachability by just enumerate all possible paths in G with length bounded by B.

7 Conclusion

We have shown that the reachability problem in d-dimensional vector addition system with states is in F_d, improving the previous F_{d+4} upper bound by Leroux and Schmitz [15]. By capturing reachability in geometrically 2-dimensional VASSes with linear path schemes, we are able to reduce significantly the number of decomposition steps in the KLMST decomposition algorithm. Combined with a careful analysis on fast-growing functions, we finally obtained the F_d upper bound. It should be noticed though, that our algorithm avoids computing the "full decomposition" [14] of KLM sequences, thus cannot improve the complexity of problems that essentially rely on the full decomposition, e.g., the VASS downward language inclusion problem [8, 23, 15].

It has been shown that the reachability problem in $(2d+3)$-VASS is F_d-hard [5]. In the case of 3-VASS, it is known that the reachability problem is PSPACE-hard. The gap between the lower bound and the upper bound $\mathsf{F}_3 = \mathsf{TOWER}$ [22] is huge. It is very unlikely that the problem is PSPACE-complete. Effort to uplift the lower bound is called for.

References

1 Michael Blondin, Matthias Englert, Alain Finkel, Stefan Göller, Christoph Haase, Ranko Lazic, Pierre McKenzie, and Patrick Totzke. The reachability problem for two-dimensional vector addition systems with states. *J. ACM*, 68(5):34:1–34:43, 2021. doi:10.1145/3464794.

2 Michael Blondin, Alain Finkel, Stefan Göller, Christoph Haase, and Pierre McKenzie. Reachability in two-dimensional vector addition systems with states is pspace-complete. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 32–43. IEEE Computer Society, 2015. doi:10.1109/LICS.2015.14.

3 Dmitry Chistikov and Christoph Haase. The taming of the semi-linear set. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 128:1–128:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.ICALP.2016.128.

4 E. A. Cichon and Elias Tahhan-Bittar. Ordinal recursive bounds for higman's theorem. *Theor. Comput. Sci.*, 201(1-2):63–84, 1998. doi:10.1016/S0304-3975(97)00009-1.

5 Wojciech Czerwinski, Ismaël Jecker, Slawomir Lasota, Jérôme Leroux, and Lukasz Orlikowski. New lower bounds for reachability in vector addition systems. In Patricia Bouyer and Srikanth Srinivasan, editors, *43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2023, December 18-20, 2023, IIIT Hyderabad, Telangana, India*, volume 284 of *LIPIcs*, pages 35:1–35:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.FSTTCS.2023.35.

6 Wojciech Czerwinski and Lukasz Orlikowski. Reachability in vector addition systems is ackermann-complete. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 1229–1240. IEEE, 2021. doi:10.1109/FOCS52979.2021.00120.

7 Christoph Haase, Stephan Kreutzer, Joël Ouaknine, and James Worrell. Reachability in succinct and parametric one-counter automata. In Mario Bravetti and Gianluigi Zavattaro, editors, *CONCUR 2009 - Concurrency Theory, 20th International Conference, CONCUR 2009, Bologna, Italy, September 1-4, 2009. Proceedings*, volume 5710 of *Lecture Notes in Computer Science*, pages 369–383. Springer, 2009. doi:10.1007/978-3-642-04081-8_25.

8 Peter Habermehl, Roland Meyer, and Harro Wimmel. The downward-closure of petri net languages. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part II*, volume 6199 of *Lecture Notes in Computer Science*, pages 466–477. Springer, 2010. doi:10.1007/978-3-642-14162-1_39.

9 Thomas Jech. *Set Theory*. Springer Monographs in Mathematics. Springer, 2003. doi:10.1007/3-540-44761-X.

10 S. Rao Kosaraju. Decidability of reachability in vector addition systems (preliminary version). In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, pages 267–281, New York, NY, USA, 1982. Association for Computing Machinery. doi:10.1145/800070.802201.

11 Jean-Luc Lambert. A structure to decide reachability in petri nets. *Theor. Comput. Sci.*, 99(1):79–104, 1992. doi:10.1016/0304-3975(92)90173-D.

12 Jérôme Leroux. Flat petri nets (invited talk). In Didier Buchs and Josep Carmona, editors, *Application and Theory of Petri Nets and Concurrency - 42nd International Conference, PETRI NETS 2021, Virtual Event, June 23-25, 2021, Proceedings*, volume 12734 of *Lecture Notes in Computer Science*, pages 17–30. Springer, 2021. doi:10.1007/978-3-030-76983-3_2.

13 Jérôme Leroux. The reachability problem for petri nets is not primitive recursive. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 1241–1252. IEEE, 2021. doi:10.1109/FOCS52979.2021.00121.

14 Jérôme Leroux and Sylvain Schmitz. 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*, pages 56–67. IEEE Computer Society, 2015. doi:10.1109/LICS.2015.16.

15 Jérôme Leroux and Sylvain Schmitz. 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*, pages 1–13. IEEE, 2019. doi:10.1109/LICS.2019.8785796.

16 Jérôme Leroux and Grégoire Sutre. On flatness for 2-dimensional vector addition systems with states. In Philippa Gardner and Nobuko Yoshida, editors, *CONCUR 2004 - Concurrency Theory, 15th International Conference, London, UK, August 31 - September 3, 2004, Proceedings*, volume 3170 of *Lecture Notes in Computer Science*, pages 402–416. Springer, 2004. doi:10.1007/978-3-540-28644-8_26.

17 Ernst W. Mayr. An algorithm for the general petri net reachability problem. In *Proceedings of the Thirteenth Annual ACM Symposium on Theory of Computing*, STOC '81, pages 238–246, New York, NY, USA, 1981. Association for Computing Machinery. doi:10.1145/800076.802477.

18 Loic Pottier. Minimal solutions of linear diophantine systems: Bounds and algorithms. In Ronald V. Book, editor, *Rewriting Techniques and Applications, 4th International Conference, RTA-91, Como, Italy, April 10-12, 1991, Proceedings*, volume 488 of *Lecture Notes in Computer Science*, pages 162–173. Springer, 1991. doi:10.1007/3-540-53904-2_94.

19 Charles Rackoff. The covering and boundedness problems for vector addition systems. *Theor. Comput. Sci.*, 6:223–231, 1978. doi:10.1016/0304-3975(78)90036-1.

20 Sylvain Schmitz. Complexity bounds for ordinal-based termination - (invited talk). In Joël Ouaknine, Igor Potapov, and James Worrell, editors, *Reachability Problems - 8th International Workshop, RP 2014, Oxford, UK, September 22-24, 2014. Proceedings*, volume 8762 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2014. doi:10.1007/978-3-319-11439-2_1.

21 Sylvain Schmitz. Complexity hierarchies beyond elementary. *ACM Trans. Comput. Theory*, 8(1):3:1–3:36, 2016. doi:10.1145/2858784.

22 Qizhe Yang and Yuxi Fu. Reachability in 3-vass is in tower. *CoRR*, abs/2306.05710, 2023. doi:10.48550/arXiv.2306.05710.

23 Georg Zetzsche. The complexity of downward closure comparisons. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 123:1–123:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.ICALP.2016.123.