

# Efficient Algorithms for Asymptotic Bounds on Termination Time in VASS

Tomáš Brázdil  
Masaryk University  
Faculty of Informatics  
Brno, Czech Republic  
brazdil@fi.muni.cz

Petr Novotný  
IST Austria  
Klosterneuburg, Austria  
petr.novotny@ist.ac.at

Krishnendu Chatterjee  
IST Austria  
Klosterneuburg, Austria  
Krishnendu.Chatterjee@ist.ac.at

Dominik Velan  
Masaryk University  
Faculty of Informatics  
Brno, Czech Republic  
xvelan1@fi.muni.cz

Antonín Kučera  
Masaryk University  
Faculty of Informatics  
Brno, Czech Republic  
kucera@fi.muni.cz

Florian Zuleger  
Forsythe, TU Wien  
Wien, Austria  
zuleger@forsythe.at

## Abstract

Vector Addition Systems with States (VASS) provide a well-known and fundamental model for the analysis of concurrent processes, parameterized systems, and are also used as abstract models of programs in resource bound analysis. In this paper we study the problem of obtaining asymptotic bounds on the termination time of a given VASS. In particular, we focus on the practically important case of obtaining polynomial bounds on termination time. Our main contributions are as follows: First, we present a polynomial-time algorithm for deciding whether a given VASS has a linear asymptotic complexity. We also show that if the complexity of a VASS is not linear, it is at least quadratic. Second, we classify VASS according to quantitative properties of their cycles. We show that certain singularities in these properties are the key reason for non-polynomial asymptotic complexity of VASS. In absence of singularities, we show that the asymptotic complexity is always polynomial and of the form  $\Theta(n^k)$ , for some integer  $k \leq d$ , where  $d$  is the dimension of the VASS. We present a polynomial-time algorithm computing the optimal  $k$ . For general VASS, the same algorithm, which is based on a complete technique for the construction of ranking functions in VASS, produces a valid lower bound, i.e., a  $k$  such that the termination complexity is  $\Omega(n^k)$ . Our results are based on new insights into the geometry of VASS dynamics, which hold the potential for further applicability to VASS analysis.

## 1 Introduction

Vector Addition Systems with States (VASS) are a fundamental model widely used in program analysis. Intuitively, a VASS consists of a finite set of control states and transitions between the control states, and a set of  $d$  counters that hold non-negative integer values, where at every transition between the control states each counter is updated by a fixed integer value. A *configuration*  $pv$  of a given

VASS is determined by the current control state  $p$  and the vector  $v$  of current counter values.

One of the most basic problems studied in program analysis is *termination* that, given a program, asks whether it always terminates. For VASS, the problem whether all paths initiated in *given* configuration reach a terminal configuration is EXPSPACE-complete. Here, a terminal configuration is a configuration where the computation is “stuck” because all outgoing transitions would decrease some counter to a negative value. The EXPSPACE-hardness follows from [28], and the upper bound from [5, 37]. Contrasting to this, the problem of *structural* VASS termination, which asks whether *all* configurations of a given VASS terminate, is solvable in polynomial time [24]. This is encouraging, because structural termination guarantees termination for all instances of the parameters represented by the counter values (i.e., all inputs, all instances of a given parameterized system, etc.).

The *quantitative* variant of the termination question asks whether a given program terminates in  $O(f(n))$  steps for every input of size  $n$ , where  $f: \mathbb{N} \rightarrow \mathbb{N}$  is some function. A significant research effort has recently been devoted to this question in the program analysis literature: Recent projects include SPEED [18, 19], COSTA [1], RAML [20], Rank [2], Loopus [33, 34], AProVE [17], CoFloCo [16], C4B [9]. The cited projects target general-purpose programming languages with the goal of designing *sound (but incomplete)* analyses that work well in practice. The question whether *sound and complete* techniques can be developed for restricted classes of programs (such as VASS), however, has received considerably less attention.

**Our contribution.** In this work, we study the quantitative variant of structural VASS termination. The *termination complexity* of a given VASS is a function  $\mathcal{L}: \mathbb{N} \rightarrow \mathbb{N} \cup \{\infty\}$  such that  $\mathcal{L}(n)$  is the length of the longest computation initiated in a configuration  $pv$  where all components of  $v$  are bounded by  $n$ . We concentrate on *polynomial* and particularly on *linear* asymptotic bounds for termination complexity, which seem most relevant for practical applications. Our main results can be summarized as follows:

**Linear bounds.** We show that the problem whether  $\mathcal{L} \in \Theta(n)$  is decidable in polynomial time. Our proof reveals that if the termination complexity is *not* linear, then it is at least quadratic (or the VASS is non-terminating). Hence, there is no VASS with asymptotic termination complexity “between”  $\Theta(n)$  and  $\Theta(n^2)$ . In addition, for strongly connected linear VASS, we compute a constant  $c \in \mathbb{Q}$  (in polynomial time) such that  $\mathcal{L}(n) = cn$  for  $n \rightarrow \infty$ . Further, a

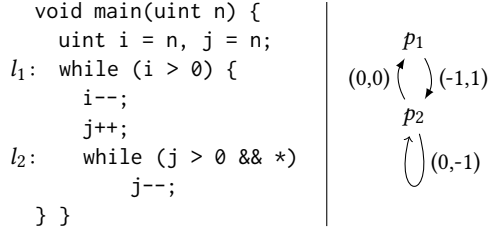
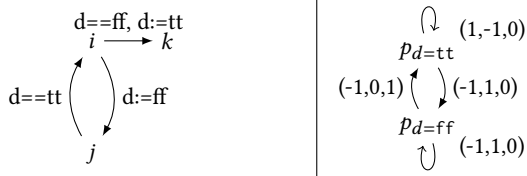
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

LICS '18, July 9–12, 2018, Oxford, United Kingdom

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5583-4/18/07...\$15.00

<https://doi.org/10.1145/3209108.3209191>

Figure 1. (a) a program, (b) VASS  $\mathcal{A}_{prog}$ Figure 2. (a) a process template, (b) VASS  $\mathcal{A}_{csys}$ 

linear VASS always has a ranking function that witnesses the linear termination complexity; this ranking function is also computable in polynomial time.

*Polynomial bounds.* We show that the termination complexity of a given VASS is highly influenced by the properties of *normals of quasi-ranking functions*, see Section 4. We start with strongly connected VASS, and classify them into the following three types:

- (A) Non-terminating VASS.
- (B) Positive normal VASS: Terminating VASS for which there exists a quasi-ranking function such that each component of its normal is positive.
- (C) Singular normal VASS: Terminating VASS for which there exists a quasi-ranking function such that each component of its normal is non-negative and (B) does not hold.

This classification is efficient, i.e., we can decide in polynomial time to which class a given VASS belongs. We show that each type (B) VASS of dimension  $d$  has termination complexity in  $\Theta(n^k)$ , where  $1 \leq k \leq d$ , and we show that the  $k$  is computable in polynomial time. Termination complexity of a type (C) VASS is not necessarily polynomial, and hence singularities in the normal are the key reason for high asymptotic bounds in VASS. For a given type (C) VASS, we show how to compute a valid lower bound, i.e., a  $k$  such that the termination complexity is  $\Omega(n^k)$  (in general, this bound does not have to be tight). Our tight analysis for type (B) VASS extends to general (not necessarily strongly connected) VASS where each SCC determines a type (B) VASS.

*Ranking Functions and Completeness.* Algorithmically the result on polynomial bounds is established by a recursive procedure: the procedure computes quasi-ranking functions which establish that certain transitions can only be taken a linear number of times; these transitions are then removed and the algorithm recurses on the remaining strongly-connected components. We show that if there is no quasi-ranking function, then the VASS does not terminate, i.e., our ranking function construction is complete. To the best of our knowledge, this is the first *completeness* result for the construction of ranking functions for VASS.

Technically, our results are based on new insights into the geometry of VASS dynamics, some of which are perhaps interesting on their own and can enrich the standard toolbox of techniques applicable to VASS analysis.

**Motivation and Illustration of our Results.** In previous work we have described automated techniques for the complexity analyses of imperative programs, which use VASS (and extensions) as backend [33, 34]. For example, our techniques allow to abstract the program given in Fig. 1 (a) to the VASS  $\mathcal{A}_{prog}$  in Fig. 1 (b).  $\mathcal{A}_{prog}$  has two locations  $p_1$  and  $p_2$ , which correspond to the loop headers of the program.  $\mathcal{A}_{prog}$  has dimension two in order to represent the variables  $i$  and  $j$ . The transitions of  $\mathcal{A}_{prog}$  correspond to the variable increments/decrements. In contrast to our previous approaches [33, 34], the analysis in this paper is guaranteed to compute tight bounds: we obtain the precise linear termination complexity  $\mathcal{L}(n) = 4n$  for  $\mathcal{A}_{prog}$  and can construct a linear ranking function, e.g.,  $f(p, (i, j)) = 3i + j + \mathbf{w}(p)$ , where  $\mathbf{w}(p_1) = 0$  and  $\mathbf{w}(p_2) = 1$  (our construction is not guaranteed to return this ranking function, but it will always find a linear ranking function).

We illustrate VASSs as models of concurrent systems: Fig. 2 (a) states a process template. A concurrent system consists of  $n$  copies of this process template. The processes communicate via the Boolean variable  $d$ . The concurrent system is equivalently represented by the VASS  $\mathcal{A}_{csys}$  in Fig. 2 (b).  $\mathcal{A}_{csys}$  has two locations  $p_{d=tt}$  and  $p_{d=ff}$ , which represent the global state.  $\mathcal{A}_{csys}$  has dimension three in order to represent the number of processes in the local states  $i, j$  and  $k$ . The transitions of  $\mathcal{A}_{csys}$  reflect the transitions of the process template, e.g., transition  $(-1, 1, 0)$  means that one process moves from state  $i$  to  $j$ . We are interested in the *parameterized verification problem*, i.e., to study the termination of the concurrent system for all system sizes  $n$ . Our results in this paper establish  $\mathcal{L}(n) \in \Theta(n^2)$ , i.e., after quadratically many steps of the concurrent system there is no more process that can take another step.

**Related Work.** *Results on VASS.* The model of VASS [22] or equivalently Petri nets are a fundamental model for parallel programs [15, 22] as well as parameterized systems [3, 4, 6]. The termination problems (counter-termination, control-state termination) as well as the related problems of boundedness and coverability have been a rich source of theoretical problems that have been widely studied [7, 13, 14, 28, 31]. The complexity of the termination problem with fixed initial configuration is EXPSPACE-complete [5, 28, 37]. Besides the termination problem, the more general reachability problem where given a VASS, an initial and a final configuration, whether there exists a path between them has also been studied [23, 25, 29]. The reachability problem is decidable [23, 25, 29], and EXPSPACE-hard [28], and the current best-known upper bound is cubic Ackermannian [26], a complexity class belonging to the third level of a fast-growing complexity hierarchy introduced in [32]. Functions (non)computable by VASS are studied in [27]. Our algorithm for computing polynomial bounds can be seen as the dual (in the sense of linear programming) of the algorithm of [24]; this connection is the basis for the completeness of our ranking function construction (we further comment on the connection to [24] in Section 4).

*Ranking functions and extensions.* Ranking functions for intraprocedural analysis have been studied widely in the literature. We restrict ourselves here to approaches which present complete methods for the construction of linear/polynomial ranking functions [2, 30, 36]; in contrast to this paper these approaches target general programs and do not show that the non-existence of a linear/polynomial ranking function implies the non-termination of the program.

The problem of existence of infinite computations in VASS has been studied in the literature. Polynomial-time algorithms have been presented in [10, 35] using results of [24]. In the more general context of games played on VASS, even deciding the existence of infinite computation is coNP-complete [10, 35], and various algorithmic approaches based on hyperplane-separation technique have been studied [11, 12, 21].

A full version of this paper is available at arXiv [8].

## 2 Preliminaries

We use  $\mathbb{N}$ ,  $\mathbb{Z}$ ,  $\mathbb{Q}$ , and  $\mathbb{R}$  to denote the sets of non-negative integers, integers, rational numbers, and real numbers. The subsets of all positive elements of  $\mathbb{N}$ ,  $\mathbb{Q}$ , and  $\mathbb{R}$  are denoted by  $\mathbb{N}^+$ ,  $\mathbb{Q}^+$ , and  $\mathbb{R}^+$ . Further, we use  $\mathbb{N}_\infty$  to denote the set  $\mathbb{N} \cup \{\infty\}$  where  $\infty$  is treated according to the standard conventions. The cardinality of a given set  $M$  is denoted by  $|M|$ . When no confusion arises, we also use  $|c|$  to denote the absolute value of a given  $c \in \mathbb{R}$ .

Given a function  $f: \mathbb{N} \rightarrow \mathbb{N}$ , we use  $O(f(n))$  and  $\Omega(f(n))$  to denote the sets of all  $g: \mathbb{N} \rightarrow \mathbb{N}$  such that  $g(n) \leq a \cdot f(n)$  and  $g(n) \geq b \cdot f(n)$  for all sufficiently large  $n \in \mathbb{N}$ , where  $a, b \in \mathbb{R}^+$  are some constants. If  $h(n) \in O(f(n))$  and  $h(n) \in \Omega(f(n))$ , we write  $h(n) \in \Theta(f(n))$ .

Let  $A, B$  be arbitrary index sets. Elements of  $\mathbb{R}^A$  are denoted by bold letters such as  $\mathbf{u}, \mathbf{v}, \mathbf{z}, \dots$ . The component of  $\mathbf{v}$  of index  $i \in A$  is denoted by  $\mathbf{v}(i)$ . For a matrix  $A \in \mathbb{R}^{A \times B}$  we denote by  $A(a, b)$  the element in row of index  $a \in A$  and column of index  $b \in B$ , and by  $A^\top$  the transpose of  $A$ . If the index set is of the form  $A = \{1, 2, \dots, d\}$  for some positive integer  $d$ , we write  $\mathbb{R}^d$  instead of  $\mathbb{R}^A$ , i.e., for  $\mathbf{v} \in \mathbb{R}^d$  we have  $\mathbf{v} = (\mathbf{v}(1), \dots, \mathbf{v}(d))$ . For every  $c \in \mathbb{N}$ , we use  $\mathbf{c}$  to denote the constant vector where all components are equal to  $c$ . The scalar product of  $\mathbf{v}, \mathbf{u} \in \mathbb{R}^d$  is denoted by  $\mathbf{v} \cdot \mathbf{u}$ , i.e.,  $\mathbf{v} \cdot \mathbf{u} = \sum_{i=1}^d \mathbf{v}(i) \cdot \mathbf{u}(i)$ . The other standard operations and relations on  $\mathbb{R}$  such as  $+$ ,  $\leq$ , or  $<$  are extended to  $\mathbb{R}^d$  in the component-wise way. In particular,  $\mathbf{v}$  is *positive* if  $\mathbf{v} > \mathbf{0}$ , i.e., all components of  $\mathbf{v}$  are positive. The norm of  $\mathbf{v}$  is defined by  $\text{norm}(\mathbf{v}) = \sqrt{\mathbf{v}(1)^2 + \dots + \mathbf{v}(d)^2}$ .

**Half-spaces and Cones.** An open half-space of  $\mathbb{R}^d$  determined by a normal vector  $\mathbf{n} \in \mathbb{R}^d$ , where  $\mathbf{n} \neq \mathbf{0}$ , is the set  $\mathcal{H}_\mathbf{n}$  of all  $\mathbf{x} \in \mathbb{R}^d$  such that  $\mathbf{x} \cdot \mathbf{n} < 0$ . A closed half-space  $\bar{\mathcal{H}}_\mathbf{n}$  is defined in the same way but the above inequality is non-strict. Given a finite set of vectors  $U \subseteq \mathbb{R}^d$ , we use  $\text{cone}(U)$  to denote the set of all vectors of the form  $\sum_{\mathbf{u} \in U} c_\mathbf{u} \mathbf{u}$ , where  $c_\mathbf{u}$  is a non-negative real constant for every  $\mathbf{u} \in U$ .

### 2.1 Syntax and semantics of VASS

In this subsection we present a syntax of VASS, represented as finite state graphs with transitions labelled by vectors of counter changes.

**Definition 2.1.** Let  $d \in \mathbb{N}$ . A  $d$ -dimensional vector addition system with states (VASS) is a pair  $\mathcal{A} = (Q, T)$ , where  $Q \neq \emptyset$  is a finite set of states and  $T \subseteq Q \times \mathbb{Z}^d \times Q$  is a finite set of transitions such that for every  $q \in Q$  there exists  $p \in Q$  and  $\mathbf{u} \in \mathbb{Z}^d$  such that  $(q, \mathbf{u}, p) \in T$ .

We denote by  $\max_{\mathcal{A}}$  the number  $\max_{(p, \mathbf{u}, q) \in T, 1 \leq i \leq d} |\mathbf{u}(i)|$ . The encoding size of  $\mathcal{A}$  is denoted by  $\|\mathcal{A}\|$  (the integers representing counter updates are written in binary).

In our discussion it is often beneficial to express constraints on transitions using matrix notation. We define the *update matrix*

$U \in \mathbb{Z}^{d \times T}$  by setting  $U(i, t) = \mathbf{u}(i)$  for all  $1 \leq i \leq d$  and all transitions  $t = (p, \mathbf{u}, p') \in T$ . We also define the *oriented incidence matrix*  $F \in \mathbb{Z}^{Q \times T}$  by setting  $F(p, t) = 1$  resp.  $F(p, t) = -1$ , if  $t = (p, \mathbf{u}, p')$  resp.  $t = (p', \mathbf{u}, p)$  and  $p' \neq p$ , and  $F(p, t) = 0$ , otherwise. We note that every column of  $F$ , corresponding to a transition  $t$ , either contains exactly one  $-1$  entry and exactly one  $1$  entry (in case the source and target of transition  $t$  are different) or only  $0$  entries (in case the source and target of transition  $t$  are the same).

**Example 2.2.** VASS  $\mathcal{A}_{\text{prog}}$  from Fig. 1 (b) has two states  $p_1, p_2$  and three transitions  $t_1 = (q_1, (-1, 1), q_2)$ ,  $t_2 = (q_2, (0, 0), q_1)$ ,  $t_3 = (q_2, (0, -1), q_2)$ . The matrices  $F$  and  $U$  look as follows:

$$F = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \end{pmatrix}$$

Here the rows correspond to the states  $q_1, q_2$  and columns to transitions  $t_1, t_2, t_3$ .

$$U = \begin{pmatrix} -1 & 0 & 0 \\ 1 & 0 & -1 \end{pmatrix}$$

Hence, the columns are the update vectors of transitions  $t_1, t_2, t_3$ .

**Paths and cycles.** A *finite path* in  $\mathcal{A}$  of length  $n$  is a finite sequence  $\pi$  of the form  $p_0, \mathbf{u}_1, p_1, \mathbf{u}_2, p_2, \dots, \mathbf{u}_n, p_n$  where  $n \geq 1$  and  $(p_i, \mathbf{u}_{i+1}, p_{i+1}) \in T$  for all  $0 \leq i < n$ . If  $p_0 = p_n$ , then  $\pi$  is a *cycle*. A cycle is *simple* if all  $p_1, \dots, p_{n-1}$  are pairwise different. The *effect* of  $\pi$ , denoted by  $\text{eff}(\pi)$ , is the sum  $\mathbf{u}_1 + \dots + \mathbf{u}_n$ . Given a set of paths  $P$ , we denote by  $\text{eff}(P)$  the sum of effects of all paths in  $P$ . Let  $\text{Inc} = \{\text{eff}(\pi) \mid \pi \text{ is a simple cycle of } \mathcal{A}\}$ . The elements of  $\text{Inc}$  are called *increments*.

Given two finite paths  $\alpha = p_0, \mathbf{u}_1, \dots, p_n$  and  $\beta = q_0, \mathbf{v}_1, \dots, q_m$  such that  $p_n = q_0$ , we use  $\alpha \odot \beta$  to denote the finite path  $p_0, \mathbf{u}_1, \dots, p_n, \mathbf{v}_1, \dots, q_m$ . A *multi-cycle* in  $\mathcal{A}$  is a multiset of simple cycles. The length of a multi-cycle is the sum of lengths of all its cycles.

Let  $\pi$  be a finite path in  $\mathcal{A}$ . A *decomposition of  $\pi$  into simple cycles*, denoted by  $\text{Decomp}(\pi)$ , is a multi-cycle, i.e., a multiset of simple cycles, defined recursively as follows:

- If  $\pi$  does not contain any simple cycle, then  $\text{Decomp}(\pi)$  is an empty multiset.
- If  $\pi = \alpha \odot \gamma \odot \beta$  where  $\gamma$  is the first simple cycle occurring in  $\pi$ , then  $\text{Decomp}(\pi) = \{\gamma\} \cup \text{Decomp}(\alpha \odot \beta)$ .

Observe that if  $\text{Decomp}(\pi)$  is empty, then the length of  $\pi$  is at most  $|Q| - 1$ . Since the length of every simple cycle is bounded by  $|Q|$ , the length of  $\pi$  is asymptotically the same as the number of elements in  $\text{Decomp}(\pi)$ , assuming a fixed VASS  $\mathcal{A}$ . Considering  $\pi'$  to be the remainder of  $\pi$  after all simple cycles of  $\text{Decomp}(\pi)$  removed by the above procedure, we obtain  $\text{eff}(\pi) = \text{eff}(\pi') + \text{eff}(\text{Decomp}(\pi))$ .

Let  $\mathcal{A} = (Q, T)$  be a VASS. A *sub-VASS* of  $\mathcal{A}$  is a VASS  $\mathcal{A}' = (Q', T')$  such that  $Q' \subseteq Q$  and  $T' \subseteq T$ . VASS  $\mathcal{A}$  is *strongly connected* if for every  $p, q \in Q$  there is a finite path from  $p$  to  $q$ .

A *strongly connected component* (SCC) of  $\mathcal{A}$  is a maximal strongly connected sub-VASS of  $\mathcal{A}$ .

**Configurations and computation.** A *configuration* of  $\mathcal{A}$  is a pair  $p\mathbf{v}$ , where  $p \in Q$  and  $\mathbf{v} \in \mathbb{N}^d$ . The set of all configurations of  $\mathcal{A}$  is denoted by  $C(\mathcal{A})$ . The size of  $p\mathbf{v} \in C(\mathcal{A})$  is defined as  $\|p\mathbf{v}\| = \|\mathbf{v}\| = \max\{\mathbf{v}(i) \mid 1 \leq i \leq d\}$ . Given  $n \in \mathbb{N}$ , we say that  $p\mathbf{v}$  is *n-bounded* if  $\|p\mathbf{v}\| \leq n$ .

A *computation* initiated in  $p_0\mathbf{v}_0$  is a finite sequence  $p_0\mathbf{v}_0, \dots, p_n\mathbf{v}_n$  of configurations such that there exists a

path  $p_0, u_1, p_1, u_2, p_2, \dots, u_n, p_n$  where  $v_i = v_0 + u_1 + \dots + u_i$  for all  $0 \leq i \leq n$ . The length of a given computation is the length of its (unique) corresponding path.

## 2.2 Termination Complexity of VASS

**Definition 2.3.** Let  $\mathcal{A} = (Q, T)$  be a  $d$ -dimensional VASS. For every configuration  $pv$  of  $\mathcal{A}$ , let  $L(pv)$  be the least  $\ell \in \mathbb{N}_\infty$  such that the length of every finite computation initiated in  $pv$  is bounded by  $\ell$ . The *termination complexity* of  $\mathcal{A}$  is a function  $\mathcal{L}: \mathbb{N} \rightarrow \mathbb{N} \cup \{\infty\}$  defined by

$$\mathcal{L}(n) = \sup \{L(pv) \mid pv \in C(\mathcal{A}) \text{ where } \|pv\| = n\}.$$

If  $\mathcal{L}(n) = \infty$  for some  $n \in \mathbb{N}$ , we say that  $\mathcal{A}$  is *non-terminating*, otherwise it is *terminating*.

Observe that if  $\mathcal{A}$  is non-terminating, then  $\mathcal{L}(n) = \infty$  for all sufficiently large  $n \in \mathbb{N}$ . Further, if  $\mathcal{A}$  is terminating, then  $\mathcal{L}(n) \in \Omega(n)$ . In particular, if  $\mathcal{L}(n) \in O(n)$ , we also have  $\mathcal{L}(n) \in \Theta(n)$ .

## 3 Linear Termination Time

In this section, we give a complete and effective characterization of all VASS with *linear* termination complexity. Let us consider a VASS  $\mathcal{A} = (Q, T)$ . We assume that  $\mathcal{A}$  is strongly connected unless explicitly stated otherwise.

Consider an integer solution  $\mu \in \mathbb{Z}^T$  to the constraints  $\mu \geq 0$  and  $F\mu = 0$  (here  $F$  is the oriented incidence matrix of  $\mathcal{A}$ ). Note that  $\mu$  induces a *multi-cycle*  $M$  of  $\mathcal{A}$ . Indeed, if  $\mu(t) > 0$ , then there is a transition  $t'$  with  $\mu(t') > 0$  such that the source state of  $t'$  is equal to the target state of  $t$ . Hence one may trace a path over states with positive value in  $\mu$  that eventually leads to a simple cycle. Subtracting one from  $\mu(t)$  for all  $t$  on the simple cycle we obtain  $\mu'$  still satisfying the above constraints. Repeating this process we eventually end up with a zero vector and the desired multi-cycle  $M$ .

Note that  $1^\top \mu$  is equal to the number of transitions traced along the multi-cycle. So, roughly speaking, it suffices to add a constraint  $U\mu \geq -n\mathbf{1}$  (here  $U$  is the update matrix) to characterize multicycles that, when appropriately executed in an  $n$ -bounded configuration, produce a zero-avoiding computation. However, there are several issues in such a formulation, namely dependency of the constraints on the parameter  $n$  and demand for an integer solution.

So we transform the constraints into the following relaxed optimization problem to completely characterize the linear computational complexity:

rational LP (R):

$$\begin{aligned} \max \quad & 1^\top \rho \\ \text{subject to} \quad & \rho \geq 0 \\ & U\rho \geq -1 \\ & F\rho = 0 \end{aligned}$$

**Theorem 3.1.** Let  $\mathcal{A}$  be a strongly connected VASS. We consider LP (R) over  $\mathbb{Q}$ .

- (A) If LP (R) has a solution  $\rho$  with  $1^\top \rho = c \in \mathbb{Q}$ , then  $cn$  is the precise asymptotic computational complexity of  $\mathcal{A}$ , i.e.,  $\mathcal{L}(n)$  converges to  $cn$  for  $n \rightarrow \infty$ .
- (B) If (R) is unbounded, then the computational complexity of  $\mathcal{A}$  is at least quadratic.

**Intuition:** Let  $\rho$  be a rational solution of (R) with  $1^\top \rho = c$  and consider a non-negative integer  $n \in \mathbb{N}$ . Let  $\mu \in \mathbb{Z}^T$  satisfy  $\mu =$

$nmp$  where  $m$  is the least common multiple of the denominators of  $\rho$ . Since  $\mu = nmp \geq 0$  and  $F\mu = Fnp = 0$ , the vector  $\mu$  specifies a multi-cycle of length  $1^\top \mu = cnm$ . Moreover,  $\mu$  satisfies  $U\mu = Unmp \geq -nm \cdot \mathbf{1}$  which means that executing all transitions of the multi-cycle cannot decrease the counters by more than  $mn$ . By executing cycles of the multi-cycle in a carefully arranged order initiated in a  $n$ -bounded configuration, we obtain a zero-avoiding computation whose length is, roughly,  $cn$ .

On the other hand, if the program (R) is unbounded, we show that then there is a solution  $\rho$  satisfying  $U\rho \geq 0$ . From this we obtain multi-cycles of arbitrary length whose overall effect is non-negative. Note that this does not mean that the VASS is non-terminating since the cycles need to be connected into a single computation. However, we show that they always can be connected into a computation of at least quadratic length.

**Proof of Theorem 3.1 (A).** Assume (R) is bounded. Let  $\rho \in \mathbb{Q}^T$  be an optimal solution. We set  $c = 1^\top \rho$ . We first show the upper bound. We fix some  $n$ . We consider the longest computation starting from some  $n$ -bounded configuration. Let  $\pi$  be the path associated to this computation. Because we are interested only in asymptotic behaviour, we can assume  $\pi$  is a cycle. Let  $\mu_\pi(t)$  denote the number of occurrences of transition  $t$  on  $\pi$ . We note that  $U\mu_\pi = \text{eff}(\pi) \geq -n \cdot \mathbf{1}$  because the starting configuration of the considered worst-case computation is  $n$ -bounded. Because  $\pi$  is a cycle, we have  $F\mu_\pi = 0$ . Hence,  $\frac{1}{n} \cdot \mu_\pi$  is a feasible point of LP (R) and we get  $1^\top \frac{1}{n} \cdot \mu_\pi \leq c$ . Thus,  $1^\top \mu_\pi \leq cn$ . Because this holds for all  $n$ , we can conclude  $\mathcal{L}(n) \leq cn$ .

We show the lower bound. We fix some  $n$ . Let  $m$  be the least common multiple of the denominators of  $\rho$ . We set  $\mu = m \cdot \rho \in \mathbb{Z}^T$ . We have  $\mu \geq 0$ ,  $U\mu \geq -m \cdot \mathbf{1}$ ,  $F\mu = 0$  and  $1^\top \mu = cm$ . We consider the multi-cycle  $M$  associated to  $\mu$ . Let  $C$  be some cycle of  $\mathcal{A}$  which visits each state at least once. Let  $l$  be the length of  $C$ . Because  $C$  visits every state at least once we can combine  $C$  and  $\sqrt{n}$  copies of multi-cycle  $M$  into a single cycle  $C'$ . Let  $l'$  be the length of  $C'$ . We have  $l' = l + \sqrt{n}1^\top \mu = l + \sqrt{n}cm$ . Let  $p$  be the start and end state of  $C'$ . We set  $n' = \frac{n - (l + \sqrt{n}cm) \cdot \max_A}{\max_A \cdot l + m\sqrt{n}}$  (rounded down if needed). Let  $v_0 = n \cdot \mathbf{1}$ . We show that starting from configuration  $pv_0$  we can  $n'$  times execute the cycle  $C'$ . This is sufficient to establish  $\mathcal{L}(n) = cn$  because of  $\frac{n'l'}{cn} \rightarrow 1$  for  $n \rightarrow \infty$ .

We consider the configurations  $pv_i$  after  $0 \leq i < n'$  executions of  $C'$ . We show by induction on  $i$  that  $C'$  can be executed one more time. We have  $\text{eff}(C') = \text{eff}(C) + \sqrt{n} \text{eff}(M) = \text{eff}(C) + \sqrt{n}U\mu \geq -(\max_A \cdot l + m\sqrt{n}) \cdot \mathbf{1}$ . Hence, we have  $v_i \geq n \cdot \mathbf{1} - i(\max_A \cdot l + m\sqrt{n}) \cdot \mathbf{1}$ . We have to show that we can execute  $C'$  one more time. In every step of  $C'$  we decrease each vector component by at most  $\max_A$ . Hence, we need to show  $v_i \geq l' \cdot \max_A \cdot \mathbf{1}$ . Indeed, we have  $v_i \geq n \cdot \mathbf{1} - i(\max_A \cdot l + m\sqrt{n}) \cdot \mathbf{1} \geq (l + \sqrt{n}cm) \cdot \max_A \cdot \mathbf{1}$ .

**Proof of Theorem 3.1 (B).** Assume (R) is unbounded. We will show that there is no open half-space  $\mathcal{H}_n$  of  $\mathbb{R}^d$  such that  $n > 0$  and  $\text{Inc} \subseteq \mathcal{H}_n$ . As we show later, this implies that the computational complexity of  $\mathcal{A}$  is at least quadratic. From the theory of linear programming we know that there is a direction in which the polyhedron given by  $\rho \geq 0$ ,  $U\rho \geq -1$  and  $F\rho = 0$  is unbounded and which increases the objective function  $1^\top \rho$ . Hence, there is a  $\rho \geq 0$  with  $U\rho \geq 0$  and  $F\rho = 0$  and  $\rho(t) \geq 1$  for some  $t \in T$ . We consider the multi-cycle  $M$  extracted from the integer vector

$\mu = m\rho$  where  $m$  is the common multiple of denominators in  $\rho$ . Assume now for the sake of contradiction that there is an open half-space  $\mathcal{H}_n$  of  $\mathbb{R}^d$  such that  $\mathbf{n} > \mathbf{0}$  and  $\text{Inc} \subseteq \mathcal{H}_n$ . Let  $C_1, \dots, C_k$  be all simple cycles occurring in  $M$ . Because of  $\text{Inc} \subseteq \mathcal{H}_n$  we have  $\mathbf{n}^\top \cdot \text{eff}(C_i) < 0$  for all  $i$ , and hence

$$0 > \sum_i \mathbf{n}^\top \text{eff}(C_i) = \mathbf{n}^\top \text{eff}(M) = \mathbf{n}^\top U\mu = \mathbf{n}^\top U m\rho = m(\mathbf{n}^\top U\rho)$$

which implies  $\mathbf{n}^\top U\rho < 0$ . On the other hand, we get  $\mathbf{n}^\top \cdot U\rho \geq 0$  from  $\mathbf{n} > \mathbf{0}$  and  $U\rho \geq \mathbf{0}$ . A contradiction.

Now suppose there is no open half-space  $\mathcal{H}_n$  of  $\mathbb{R}^d$  such that  $\mathbf{n} > \mathbf{0}$  and  $\text{Inc} \subseteq \mathcal{H}_n$ . We show that  $\mathcal{L}(n) \in \Omega(n^2)$ , i.e., there exist  $p \in Q$  and a constant  $a \in \mathbb{R}^+$  such that for all configurations  $p\mathbf{n}$ , where  $n \in \mathbb{N}$  is sufficiently large, there is a computation initiated in  $p\mathbf{n}$  whose length is at least  $a \cdot n^2$ .

The crucial point is that now there are  $\mathbf{v}_1, \dots, \mathbf{v}_k \in \text{Inc}$  and  $b_1, \dots, b_k \in \mathbb{N}^+$  such that  $k \geq 1$  and  $\sum_{i=1}^k b_i \mathbf{v}_i \geq \mathbf{0}$ .

The above is a direct consequence of the following purely geometric lemma with  $X = \text{Inc}_{\mathcal{A}}$ .

**Lemma 3.2.** *Let  $X \subset \mathbb{Z}^d$  be a finite set. If there is no open half-space  $\mathcal{H}_n$  of  $\mathbb{R}^d$  such that  $\mathbf{n} > \mathbf{0}$  and  $X \subseteq \mathcal{H}_n$ , then there exist  $\mathbf{v}_1, \dots, \mathbf{v}_k \in X$  and  $b_1, \dots, b_k \in \mathbb{N}^+$  such that  $k \geq 1$  and  $\sum_{i=1}^k b_i \mathbf{v}_i \geq \mathbf{0}$ .*

As the individual simple cycles with effects  $\mathbf{v}_1, \dots, \mathbf{v}_k$  may proceed through disjoint sets of states, they *cannot* be trivially concatenated into one large cycle with non-negative effect. Instead, we fix a control state  $p \in Q$  and a cycle  $\pi$  initiated in  $p$  visiting *all* states of  $Q$ . Further, for every  $1 \leq i \leq k$  we fix a simple cycle  $\gamma_i$  such that  $\text{eff}(\gamma_i) = \mathbf{v}_i$ . For every  $t \in \mathbb{N}$ , let  $\pi_t$  be a cycle obtained from  $\pi$  by inserting precisely  $t \cdot b_i$  copies of every  $\gamma_i$ , where  $1 \leq i \leq k$ . Observe that the inequality  $\sum_{i=1}^k b_i \mathbf{v}_i \geq \mathbf{0}$  implies

$$\text{eff}(\pi_t) = \text{eff}(\pi) + t \cdot \sum_{i=1}^k b_i \mathbf{v}_i \geq \text{eff}(\pi) \quad \text{for every } t \in \mathbb{N}. \quad (1)$$

For every configuration  $p\mathbf{u}$ , let  $t(\mathbf{u})$  be the largest  $t \in \mathbb{N}$  such that  $\pi_t$  is executable in  $p\mathbf{u}$ . If such a  $t(\mathbf{u})$  does not exist, i.e.  $\pi_t$  is executable in  $p\mathbf{u}$  for all  $t \in \mathbb{N}$ , then  $\mathcal{A}$  is non-terminating (since, e.g.  $\mathbf{v}_1$  must be non-negative in such a case), and the proof is finished. Hence, we can assume that  $t(\mathbf{u})$  is well-defined for each  $\mathbf{u}$ . Since the cycles  $\pi$  and  $\gamma_1, \dots, \gamma_k$  have fixed effects, there is  $b \in \mathbb{R}^+$  such that for all configurations  $p\mathbf{u}$  where all components of  $\mathbf{u}$  are above some sufficiently large threshold  $\xi$  we have that  $t(\mathbf{u}) \geq b \cdot \min_{j=1, \dots, d} u_j$ , i.e.  $t(\mathbf{u})$  grows asymptotically at least linearly with the minimal component of  $\mathbf{u}$ . Now, for every  $n \in \mathbb{N}$ , consider a computation  $\alpha(n)$  initiated in  $p\mathbf{n}$  defined inductively as follows: Initially,  $\alpha(n)$  consists just of  $p\mathbf{u}_0 = p\mathbf{n}$ ; if the prefix of  $\alpha(n)$  constructed so far ends in a configuration  $p\mathbf{u}_i$  such that  $t(\mathbf{u}_i) \geq 1$  and  $\mathbf{u}_i \geq \xi$  (an event we call a *successful hit*), then the prefix is prolonged by executing the cycle  $\pi_{t(\mathbf{u}_i)}$  (otherwise, the construction of  $\alpha(n)$  stops). Thus,  $\alpha(n)$  is obtained from  $p\mathbf{n}$  by applying the inductive rule  $I(n)$  times, where  $I(n) \in \mathbb{N}_\infty$  is the number of successful hits before the construction of  $\alpha(n)$  stops. Denote by  $p\mathbf{u}_i$  the configuration visited by  $\alpha(n)$  at  $i$ -th successful hit. Now the inequality (1) implies that  $\mathbf{u}_i \geq \mathbf{n} + i \cdot \text{eff}(\pi)$ , so there exists a constant  $e$  such that  $\min_{j=1, \dots, d} u_i(j) \geq n - i \cdot e$ . In particular the decrease of all components of  $\mathbf{u}_i$  is at most linear in  $i$ . This means that  $I(n) \geq c \cdot n$  for all sufficiently large  $n \in \mathbb{N}$ , where  $c \in \mathbb{R}^+$  is a suitable constant. But at the same time, upon each successful

$$\begin{array}{l} \min \mathbf{y}_U^\top \cdot \mathbf{1} \text{ with} \\ \mathbf{y}_U^\top \cdot U - \mathbf{y}_F^\top \cdot F \leq -\mathbf{1}^\top \\ \mathbf{y}_U \geq \mathbf{0} \end{array}$$

**Figure 3.** The rational LP  $R_{dual}$  that is dual to  $R$ . Here the variables are vectors  $\mathbf{y}_U \in \mathbb{Q}^d$  and  $\mathbf{y}_F \in \mathbb{Q}^Q$ .

hit we have  $\mathbf{u}_i \geq \xi$ , so the length of the segment beginning with the  $i$ -th successful hit and ending with the  $(i+1)$ -th hit or with the last configuration of  $\alpha(n)$  is at least  $b \cdot \min_{j=1, \dots, d} u_i(j) \geq b \cdot (n - i \cdot e)$ . Hence, the length of  $\alpha(n)$  is at least  $\sum_{i=1}^{c \cdot n} b \cdot (n - i \cdot e)$ , i.e., quadratic.

Finally, let us consider an arbitrary VASS  $\mathcal{A}$ , not necessarily strongly connected. The following lemma allows us to characterize the linear complexity of termination for  $\mathcal{A}$  by applying Theorem 3.1 to its strongly connected components. A proof is straightforward.

**Lemma 3.3.** *Let  $d \in \mathbb{N}$ , and let  $\mathcal{A} = (Q, T)$  be a  $d$ -dimensional VASS. Then  $\mathcal{L}(n) \in O(n)$  iff  $\mathcal{L}_R(n) \in O(n)$  for every SCC  $R$  of  $Q$ , where  $\mathcal{L}_R(n)$  is the termination complexity of  $\mathcal{A}_R$ .*

**Corollary 3.4.** *The problem whether the termination complexity of a given  $d$ -dimensional VASS is linear is solvable in time polynomial in the size of  $\mathcal{A}$ .*

## 4 Polynomial termination time

We now concentrate on VASS with polynomial termination complexity. For simplicity, we restrict ourselves to *strongly connected* VASS. The general case is discussed at the end of the section.

A prominent notion in our analysis is the one of a *ranking function* for VASS. Let  $\mathcal{A} = (Q, T)$  be a VASS. A *linear map* for  $\mathcal{A}$  is a function  $f$  assigning rational numbers to configurations of  $\mathcal{A}$  s.t. there exists a vector  $\mathbf{c}_f$  and a weighting vector  $\mathbf{w}_f \in \mathbb{Q}^Q$  such that for each configuration  $p\mathbf{v}$  of  $\mathcal{A}$  it holds  $f(p\mathbf{v}) = \mathbf{c}_f^\top \cdot \mathbf{v} + \mathbf{w}_f(p)$ . The vector  $\mathbf{c}_f$  is called a *normal* of  $f$ . Given a linear map  $f$ , we say that a transition  $(p, \mathbf{u}, q)$  of  $\mathcal{A}$  is *f-ranked* if  $\mathbf{c}_f^\top \cdot \mathbf{u} + \mathbf{w}_f(q) \leq \mathbf{c}_f^\top \cdot \mathbf{v} + \mathbf{w}_f(p) - 1$  and *f-neutral* if  $\mathbf{c}_f^\top \cdot \mathbf{u} + \mathbf{w}_f(q) = \mathbf{c}_f^\top \cdot \mathbf{v} + \mathbf{w}_f(p)$ . A linear map  $f$  is a *quasi-ranking function* (QRF) for  $\mathcal{A}$  if  $\mathbf{c}_f \geq \mathbf{0}$  and if all transitions of  $\mathcal{A}$  are either *f-ranked* or *f-neutral*, and a *ranking function* (RF) if  $\mathbf{c}_f \geq \mathbf{0}$  and all transitions of  $\mathcal{A}$  are *f-ranked*. A quasi-ranking function  $f$  is *positive* if each component of  $\mathbf{c}_f$  is positive. Note that in the language of update and incidence matrices  $U$  and  $F$  the conditions can be phrased as follows: a linear map  $f$  is a QRF if and only if  $\mathbf{c}_f \geq \mathbf{0}$  and  $\mathbf{c}_f^\top \cdot U - \mathbf{w}_f^\top \cdot F \leq \mathbf{0}^\top$  such that if there is a negative number in some column, it is  $\leq -1$ . Similarly, a linear map  $f$  is a RF if and only if  $\mathbf{c}_f \geq \mathbf{0}$  and  $\mathbf{c}_f^\top \cdot U - \mathbf{w}_f^\top \cdot F \leq -\mathbf{1}^\top$ .

The existence of ranking functions is already tightly connected to the question whether a given VASS has linear complexity, as shown in the following theorem.

**Theorem 4.1.** *A VASS  $\mathcal{A}$  has a linear termination complexity if and only if there exists a ranking function for  $\mathcal{A}$ .*

*Proof.* Consider the LP  $R$  from Theorem 3.1. Its dual LP is the LP  $R_{dual}$  pictured in Figure 3.

The dual LP has a feasible solution if and only if the original LP has an optimal solution (since it always has a feasible solution)

and that is if and only if the VASS  $\mathcal{A}$  is linear (due to Theorem 3.1). Assume there exists a feasible solution. Let  $f$  be a function such that

$$f(pv) = y_U^\top \cdot v + y_F(p)$$

i.e.,  $c_f = y_U$  and  $w_f(p) = y_F(p)$ . From the constraints of the dual LP we obtain for any transition  $(p, u, q)$

$$c_f^\top \cdot u - (w_f(p) - w_f(q)) \leq -1,$$

i.e.  $f$  is a RF. Conversely, let  $f$  be any RF. Then  $y_U = c_f$ ,  $y_F = w_f$  is a feasible solution for the dual LP.  $\square$

Below, we show that complexity of general VASS  $\mathcal{A}$  is highly influenced by properties of normals of QRFs for  $\mathcal{A}$ . In particular, we classify each VASS  $\mathcal{A}$  into one of three types:

- (A) Non-terminating VASS.
- (B) Positive normal VASS: Terminating VASS  $\mathcal{A}$  for which there exists a QRF  $f$  s.t. each component of the normal  $c_f$  is positive.
- (C) Singular normal VASS: Terminating VASS  $\mathcal{A}$  for which there exists a QRF  $f$  for  $\mathcal{A}$  s.t. each the normal  $c_f$  is non-negative and (B) does not hold.

**Results.** We perform our complexity analysis on top of the above classification. We show that each non-trivial type (B) VASS of dimension  $d$  has termination complexity in  $\Theta(n^k)$ , where  $1 \leq k \leq d$  is an integer. Condition (C) is not strong enough to guarantee polynomial termination complexity, and hence singularities in the QRF normals are the key reason for complex asymptotic bounds in VASS. On the algorithmic front, we present a polynomial-time algorithm which classifies VASS into one of the above classes. Moreover, for type (B) VASS the algorithm also computes the degree  $k$  such that the termination complexity of the VASS is  $\Theta(n^k)$ . Hence, we give a complete complexity classification of type (B) VASS. For type (C) VASS, the algorithm returns a valid lower bound: a  $k$  such that the termination complexity is  $\Omega(n^k)$  (in general, such a bound does not have to be tight). In the following, we first present the algorithm and then formally state and prove its properties, which establish the above results.

Theorem 4.1 gives complete classification of linear complexity VASS. Note that the ranking function doesn't have to be positive. The following lemma shows that every linear VASS is actually of type (B).

**Lemma 4.2.** *Let  $\mathcal{A}$  be a VASS. There exists a ranking function for  $\mathcal{A}$  if and only if there exists a positive ranking function for  $\mathcal{A}$ .*

*Proof.* One direction is trivial. For the other, assume we have some ranking function  $f$  for  $\mathcal{A}$ . Then for any transition  $t = (p, u, q)$  we have  $c_f \cdot u + w_f(q) \leq w_f(p) - 1$ .

Let  $\epsilon > 0$  be such that every transition  $(p, u, q)$  we have  $\epsilon \cdot u \leq 1$  (there are only finitely many transitions so such  $\epsilon$  must exist). We define a linear map  $g$  as follows

$$c_g = 2c_f + \epsilon \quad \text{and} \quad w_g = 2w_f.$$

Then for any transition  $(p, u, q)$  we have

$$\begin{aligned} c_g \cdot u + w_g(q) &= 2c_f \cdot u + \epsilon \cdot u + 2w_f(q) \leq \\ &2w_f(p) - 2 + \epsilon \cdot u \leq 2w_f(p) - 1 = w_g(p) - 1. \end{aligned}$$

Therefore,  $g$  is a positive RF.  $\square$

---

**Algorithm 1:** Computing polynomial upper/lower bounds on the termination complexity of  $\mathcal{A}$ .

---

**input** : A strongly connected  $d$ -dimensional VASS  $\mathcal{A} = (Q, T)$  with at least one transition.  
**output** : A tuple  $(k, \text{tight}) \in \{1, 2, \dots, d\} \times \{\text{true}, \text{false}\}$ , or “non-terminating”.

```

1 if  $\exists$  positive QRF for  $\mathcal{A}$  then  $\text{tight} \leftarrow \text{true}$ 
2 else  $\text{tight} \leftarrow \text{false}$ 
3  $k := \text{Decompose}(\mathcal{A})$ 
4 if  $k = \infty$  then return “non-terminating”
5 else return  $(k, \text{tight})$ 

6 procedure  $\text{Decompose}(\mathcal{A})$ 
7    $f \leftarrow$  a QRF for  $\mathcal{A}$  maximizing the no. of  $f$ -ranked transitions
8    $T_f \leftarrow \{f\text{-neutral transitions of } \mathcal{A}\}$ 
9   if  $T_f$  contains all transitions of  $\mathcal{A}$  then return  $\infty$ 
10  if  $T_f = \emptyset$  then return 1
11   $\mathcal{A}_1, \dots, \mathcal{A}_\ell \leftarrow$  all SCCs of  $\mathcal{A}_{T_f}$ 
12  return  $1 + \max(\text{Decompose}(\mathcal{A}_1), \dots, \text{Decompose}(\mathcal{A}_\ell))$ 
```

---

**Algorithm.** Our method is formalized in Algorithm 1. In the algorithm, for a VASS  $\mathcal{A} = (Q, T)$  and  $T' \subset T$ , we denote by  $\mathcal{A}_{T'} = (Q, T')$  a pair obtained from  $\mathcal{A}$  by removing all transitions not belonging to  $T'$ . Note that this may not be a VASS (since some state doesn't have to have an outgoing transition). An SCC of  $\mathcal{A}_{T'}$  is a maximal strongly connected VASS in  $\mathcal{A}_{T'}$ . We now formally state the properties of the algorithm, starting with bounds on its running time.

**Theorem 4.3.** *Algorithm 1 runs in time polynomial in  $\|\mathcal{A}\|$ . In particular, when called on a VASS of dimension  $d$ , the overall depth of recursion is  $< d$ .*

We proceed with correctness of the algorithm w.r.t. non-termination.

**Theorem 4.4.** *Assume that on input  $\mathcal{A}$ , Algorithm 1 returns “non-terminating.” Then  $\mathcal{A}$  is a non-terminating VASS.*

Finally, the following two theorems show the correctness of the algorithm w.r.t. upper and lower bounds on the termination complexity of VASS.

**Theorem 4.5.** *Assume that on input  $\mathcal{A}$ , Algorithm 1 returns a tuple  $(k, \text{tight}) \in \mathbb{N} \times \{\text{true}, \text{false}\}$ . Then  $k \in \{1, \dots, d\}$  and  $\mathcal{A}$  is terminating. Moreover, if  $\text{tight} = \text{true}$ , then  $\mathcal{L}(n) \in O(n^k)$ .*

**Theorem 4.6.** *Assume that on input  $\mathcal{A}$ , Algorithm 1 returns a tuple  $(k, \text{tight}) \in \mathbb{N} \times \{\text{true}, \text{false}\}$ . Then  $k \in \{1, \dots, d\}$  and  $\mathcal{L}(n) \in \Omega(n^k)$ .*

Note that the algorithm indeed performs the required classification since  $\text{tight}$  is set to  $\text{true}$  if and only if the check for the existence of a positive QRF in the beginning of the algorithm is successful. We now present the proofs of the above theorems.

**Proof of Theorem 4.3.** In order to analyze the termination of the algorithm we consider the cone of cycle effects. As usual we define the dimension  $\dim(C)$  of a cone  $C$  as the dimension of the smallest vector space containing  $C$ . We show that the dimension of the cone generated by  $\text{Inc}_{\mathcal{A}}$  decreases with each recursive call:

**Lemma 4.7.** *Let  $\mathcal{A}$  be some VASS such that  $\text{Decompose}(\mathcal{A})$  leads to some recursive call  $\text{Decompose}(\mathcal{A}')$  for some SCC  $\mathcal{A}'$  of  $\mathcal{A}$ . Then  $\dim(\text{cone}(\text{Inc}_{\mathcal{A}})) > \dim(\text{cone}(\text{Inc}_{\mathcal{A}'}))$ .*

*Proof.* Clearly,  $\text{Inc}_{\mathcal{A}'} \subseteq \text{Inc}_{\mathcal{A}}$  because  $\mathcal{A}'$  is a sub-VASS of  $\mathcal{A}$  and hence  $\dim(\text{cone}(\text{Inc}_{\mathcal{A}})) \geq \dim(\text{cone}(\text{Inc}_{\mathcal{A}'}))$ . Let  $f$  be the QRF computed for  $\mathcal{A}$  and let  $\mathbf{c}_f$  be the associated normal. We will show that  $\text{cone}(\text{Inc}_{\mathcal{A}'})$  is contained in the hyperplane  $\{\mathbf{v} \mid \mathbf{c}_f^\top \mathbf{v} = 0\}$  while  $\text{cone}(\text{Inc}_{\mathcal{A}})$  is not. This is sufficient to infer  $\dim(\text{cone}(\text{Inc}_{\mathcal{A}})) > \dim(\text{cone}(\text{Inc}_{\mathcal{A}'}))$ .

We consider some  $\mathbf{v} \in \text{Inc}_{\mathcal{A}'}$ . We have  $\text{eff}(C) = \mathbf{v}$  for some simple cycle  $C$ . We consider the edges along the cycle  $C$ . Because  $\mathcal{A}'$  appears in some recursive call of  $\text{Decompose}(\mathcal{A})$ , we have that every edge of  $C$  is  $f$ -neutral, i.e.,  $\mathbf{c}_f^\top \mathbf{u} + \mathbf{w}_f(p') = \mathbf{w}_f(p)$  for every transition  $(p, \mathbf{u}, p')$  of  $C$ . Adding these equations along the cycle  $C$  establishes  $\mathbf{v} = \text{eff}(C) = 0$ . Hence,  $\text{cone}(\text{Inc}_{\mathcal{A}'}) \subseteq \{\mathbf{v} \mid \mathbf{c}_f^\top \mathbf{v} = 0\}$ .

On the other hand, because there is a recursive call i.e.,  $T_f \neq T$ , there is at least one transition of  $\mathcal{A}$  which is  $f$ -ranked. Because  $\mathcal{A}$  is connected we can choose some simple cycle  $C$  which contains an  $f$ -ranked transition. Adding the inequalities  $\mathbf{c}_f^\top \mathbf{u} + \mathbf{w}_f(p') \leq \mathbf{w}_f(p)$  for every transition  $(p, \mathbf{u}, p')$  of  $C$  establishes  $\mathbf{c}_f^\top \text{eff}(C) < 0$ . Hence,  $\text{cone}(\text{Inc}_{\mathcal{A}})$  is not contained in  $\{\mathbf{v} \mid \mathbf{c}_f^\top \mathbf{v} = 0\}$ .  $\square$

By Lemma 4.7 we have that the dimension of  $\text{cone}(\text{Inc}_{\mathcal{A}})$  decreases with every recursive call. With  $\dim(\text{cone}(\text{Inc}_{\mathcal{A}})) \leq d$ , we get that the recursion depth is bounded by  $d - 1$ .

Now we focus on the complexity of computing a QRF  $f$  maximizing the number of  $f$ -ranked transitions. The computation of such a QRF can be directly encoded by the following linear optimization problem  $Q$ .

$$\begin{aligned} \text{LP } (Q): \quad & \max \mathbf{1}^\top \mathbf{b} \\ & \mathbf{0} \leq \mathbf{b} \leq \mathbf{1} \\ & \mathbf{c} \geq \mathbf{0} \\ & \mathbf{c}^\top \cdot U - \mathbf{w}^\top \cdot F \leq -\mathbf{b}^\top \end{aligned}$$

**Lemma 4.8.** *Let  $\mathbf{c}, \mathbf{w}, \mathbf{b}$  be an optimal solution to LP  $(Q)$ . Then,  $f(pv) = \mathbf{c}^\top \mathbf{v} + \mathbf{w}(p)$  is a QRF, which is maximizing the number of  $f$ -ranked transitions.*

*Proof.* We state the following properties about LP  $(Q)$ , which are easy to verify: LP  $(Q)$  is always satisfiable (consider  $\mathbf{c} = \mathbf{0}$ ,  $\mathbf{w} = \mathbf{0}$  and  $\mathbf{b} = \mathbf{0}$ ). Let  $\mathbf{c}, \mathbf{w}, \mathbf{b}$  and  $\mathbf{c}', \mathbf{w}', \mathbf{b}'$  be feasible points of LP  $(Q)$ . Then, (1)  $\mathbf{c} + \mathbf{c}', \mathbf{w} + \mathbf{w}', \max\{\mathbf{b}, \mathbf{b}'\}$  is a feasible point of LP  $(Q)$  and (2)  $d\mathbf{c}, d\mathbf{w}, \min\{d\mathbf{b}, \mathbf{1}\}$  is a feasible point of LP  $(Q)$  for all  $d \in \mathbb{Q}$  with  $d \geq 1$ .

We now show that every transition is either  $f$ -ranked or  $f$ -neutral. It is sufficient to show that  $\mathbf{b}(t) = 0$  or  $\mathbf{b}(t) = 1$  for each transition  $t$ . Assume  $0 < \mathbf{b}(t) < 1$  for some transition  $t$ . Then, we can choose  $d = \frac{1}{\mathbf{b}(t)} > 1$  and apply (2) in order to obtain a feasible point with value  $\mathbf{1}^\top d\mathbf{b} > \mathbf{1}^\top \mathbf{b}$ , which is a contradiction to the assumption that  $\mathbf{c}, \mathbf{w}, \mathbf{b}$  is optimal.

Assume that there is another QRF  $f'(pv) = \mathbf{c}'^\top \mathbf{v} + \mathbf{w}'(p)$  and a transition  $t$  such that  $t$  is  $f'$ -ranked but  $f$ -neutral (we note that we must have  $\mathbf{b}(t) = 0$ ). We set  $\mathbf{b}'(t) = 1$  and  $\mathbf{b}'(t') = 0$  for all transitions  $t' \neq t$ . We can now apply (1) in order to obtain a feasible point with value  $\mathbf{1}^\top \max\{\mathbf{b}, \mathbf{b}'\} > \mathbf{1}^\top \mathbf{b}$ , which is a contradiction to the assumption that  $\mathbf{c}, \mathbf{w}, \mathbf{b}$  is optimal.  $\square$

Similarly, checking the existence of a positive QRF can be performed by a direct reduction to linear programming. The LP is analogous to  $Q$ .

**Lemma 4.9.** *Checking the existence of a positive QRF can be done in polynomial time.*

*Proof.* Consider the following LP similar to the dual LP in the proof of Theorem 4.1.

$$\begin{aligned} \max \varepsilon \text{ with} \\ \mathbf{y}_U^\top \cdot U - \mathbf{y}_F^\top \cdot F \leq \mathbf{0}^\top \\ \mathbf{y}_U \geq \varepsilon \end{aligned}$$

Here the variables are  $\mathbf{y}_U, \mathbf{y}_F$  and  $\varepsilon$ . We show that this program has a feasible positive solution if and only if there exists a positive QRF.

If the program has a feasible positive solution then define a linear map  $g$  such that  $\mathbf{c}_g = \mathbf{y}_U$  and  $\mathbf{w}_g = \mathbf{y}_F$ . For every transition  $(p, \mathbf{u}, q)$  we have

$$\mathbf{c}_g^\top \cdot \mathbf{u} - (\mathbf{w}_g(p) - \mathbf{w}_g(q)) \leq 0.$$

Every transition  $t = (p, \mathbf{u}, q)$  that is not  $g$ -neutral, we have

$$\mathbf{c}_g^\top \cdot \mathbf{u} - (\mathbf{w}_g(p) - \mathbf{w}_g(q)) = -\delta_t.$$

Let  $\delta = \min_{t \in T} \delta_t$ . Then function  $f = \frac{g}{\delta}$  is a positive QRF.

Positive QRF is a feasible solution for the dual program of Theorem 4.1. It is therefore a feasible positive solution for this LP (since these constraints are weaker).  $\square$

We now finish the proof of Theorem 4.3. We note that computing the QRFs in the algorithm can be done by linear programming. We next consider the set of recursive calls made at recursion depth  $i$ . The VASSs of these recursive calls are all disjoint sub-VASSs of  $\mathcal{A}$ . Thus, the complexity of solving all the optimization problems at level  $i$  is bounded by the complexity of solving  $Q$  for  $\mathcal{A}$ . Hence, the overall complexity of  $\text{Decompose}(\mathcal{A})$  is the complexity of solving  $Q$  times the dimension  $d$ .

**Proof of Theorem 4.4.** Let  $\mathcal{A} = (Q, T)$  be a VASS. Consider the constraint systems  $(A_t)$  and  $(B_t)$  stated below. Both constraint systems are parameterized by a transition  $t \in T$ . Constraint system  $(A_t)$  is taken from Kosaraju and Sullivan [24]. Note that system  $(A_t)$  has a rational solution if and only if it has an integer solution.

constraint system $(A_t)$ :	constraint system $(B_t)$ :
$U\boldsymbol{\mu} \geq \mathbf{0}$ (2)	$\mathbf{c} \geq \mathbf{0}$
$\boldsymbol{\mu} \geq \mathbf{0}$ (3)	$\mathbf{c}^\top \cdot U - \mathbf{w}^\top \cdot F \leq \mathbf{0}^\top$ with $-1$
$F\boldsymbol{\mu} = \mathbf{0}$ (4)	in column $t$
$\boldsymbol{\mu}(t) \geq 1$ (5)	

The next lemma shows the connection between  $(A_t)$  and multi-cycles in  $\mathcal{A}$ . We call a multi-cycle  $M$  non-negative if  $\text{eff}(M) \geq \mathbf{0}$ .

**Lemma 4.10** (Cited from [24]). *There is a solution  $\boldsymbol{\mu} \in \mathbb{Z}^T$  to constraints (2)-(4) iff there exists a non-negative multi-cycle  $M$  such that the number of times a transition  $t$  appears in cycles of  $M$  is at least  $\boldsymbol{\mu}(t)$ , for each  $t \in T$ .*

On the other hand, the system  $(B_t)$  is connected to QRFs.

**Lemma 4.11.** *Constraint system  $(B_t)$  has a rational solution  $\mathbf{c}, \mathbf{w}$  if and only if there exists a  $k \in \mathbb{R}^+$  and a QRF  $f$  with  $\mathbf{c}_f = k \cdot \mathbf{c}$  and  $\mathbf{w}_f = k \cdot \mathbf{w}$  such that transition  $t$  is  $f$ -ranked and every other transition is  $f$ -ranked or  $f$ -neutral.*

*Proof.* We know that a linear map  $f$  is a QRF if and only if  $\mathbf{c}_f \geq 0$  and  $\mathbf{c}_f^\top \cdot U - \mathbf{w}_f^\top \cdot F \leq \mathbf{0}^\top$  such that each column contains either 0 or a number  $\leq -1$ . Transitions with column  $\leq -1$  are  $f$ -ranked, the rest are  $f$ -neutral. Therefore, any QRF  $f$  with  $(\mathbf{c}_f^\top \cdot U)(t) - (\mathbf{w}_f^\top \cdot F)(t) \leq -1$  satisfies the constraints.

Now let  $\mathbf{c}, \mathbf{w}$  be a rational solution of  $(B_t)$ . Let

$$\mathbf{u}^T = \mathbf{c}^\top \cdot U - \mathbf{w}^\top \cdot F \leq \mathbf{0}^T.$$

Surely  $\mathbf{u}(t) \leq -1$ . But it may happen that  $0 > \mathbf{u}(t') > -1$  for some  $t' \in T$ . To remedy this, consider  $k = \max_{t' \in T} \mathbf{u}(t')$ . Let

$$\mathbf{c}_f = \frac{1}{|k|} \mathbf{c}, \quad \mathbf{w}_f = \frac{1}{|k|} \mathbf{w}.$$

Now if  $\mathbf{u}(t') < 0$ , then  $\frac{1}{|k|} \mathbf{u}(t') \leq -1$ . Therefore,  $f$  is a QRF such that  $t$  is  $f$ -ranked.  $\square$

The following result is an immediate consequence of Farkas' lemma.

**Lemma 4.12.** *For each  $t \in T$  exactly one of the constraint systems  $(A_t)$  and  $(B_t)$  has a solution.*

*Proof.* We will use the following variant of Farkas' Lemma, which states that given matrices  $A, C$  and vectors  $\mathbf{b}, \mathbf{d}$ , exactly one of the following statements is true:

constraint system (A): there exists $\mathbf{x}$ with	constraint system (B): there exist $\mathbf{y}, \mathbf{z}$ with
$A\mathbf{x} \geq \mathbf{b}$	$\mathbf{y} \geq \mathbf{0}$
$C\mathbf{x} = \mathbf{d}$	$\mathbf{y}^\top \cdot A + \mathbf{z}^\top \cdot C = \mathbf{0}^\top$
	$\mathbf{y}^\top \cdot \mathbf{b} + \mathbf{z}^\top \cdot \mathbf{d} > 0$

We fix some transition  $t$ . We denote by  $\text{char}_t \in \mathbb{Z}^Q$  the vector with  $\text{char}_t(t') = 1$ , if  $t' = t$ , and  $\text{char}_t(t') = 0$ , otherwise. Using this notation we rewrite  $(A_t)$  to the equivalent constraint system  $(A'_t)$ , where  $\text{Id}$  denotes the identity matrix:

constraint system $(A'_t)$ :	$\begin{pmatrix} U \\ \text{Id} \end{pmatrix} \mu \geq \begin{pmatrix} \mathbf{0} \\ \text{char}_t \end{pmatrix}$
	$F\mu = \mathbf{0}$

Using Farkas' Lemma (note that  $\mathbf{z}$  is not restricted so we can take  $\mathbf{z} = -\mathbf{w}$ ), we see that either  $(A'_t)$  is satisfiable or the following constraint system  $(B'_t)$  is satisfiable:

constraint system $(B'_t)$ :	simplified version of constraint system $(B'_t)$ :
$\begin{pmatrix} \mathbf{c} \\ \mathbf{y} \end{pmatrix} \geq \mathbf{0}$	$\mathbf{c} \geq \mathbf{0}$
$\begin{pmatrix} \mathbf{c} \\ \mathbf{y} \end{pmatrix}^\top \cdot \begin{pmatrix} U \\ \text{Id} \end{pmatrix} - \mathbf{w}^\top \cdot F = \mathbf{0}^\top$	$\mathbf{y} \geq \mathbf{0}$
$\begin{pmatrix} \mathbf{c} \\ \mathbf{y} \end{pmatrix}^\top \cdot \begin{pmatrix} \mathbf{0} \\ \text{char}_t \end{pmatrix} - \mathbf{w}^\top \cdot \mathbf{0} > 0$	$\mathbf{c}^\top \cdot U + \mathbf{y}^\top - \mathbf{w}^\top \cdot F = \mathbf{0}^\top$
	$\mathbf{y}(t) > 0$

We recognize that constraint systems  $(B'_t)$  and  $(B_t)$  are equivalent, because solutions of  $(B'_t)$  with  $\mathbf{y}(t) > 0$  can always be turned

into solutions with  $\mathbf{y}(t) \geq 1$  by multiplying with a sufficiently large positive rational number.  $\square$

We now finish the proof of Theorem 4.4. Because Algorithm 1 returns “non-terminating”, there is a sub-VASS  $\mathcal{A}'$  of  $\mathcal{A}$ , encountered during some recursive call, such that no transition of  $\mathcal{A}$  is  $f$ -ranked for any QRF  $f$ . Hence, constraint system  $(B_t)$  is unsatisfiable for every transition  $t$  of  $\mathcal{A}'$ . By Lemma 4.12, constraint system  $(A_t)$  is satisfiable. We consider the non-negative multi-cycle  $M$  associated to an integer solution of  $(A_t)$ . This multi-cycle contains at least transition  $t$ . Because such a multi-cycle exists for every transition  $t$ , we can combine all these multi-cycles into a single non-negative cycle, which shows that  $\mathcal{A}$  is non-terminating.

**Connection to [24].** Algorithm 1 extends algorithm ZCYCLE of Kosaraju & Sullivan [24] by a ranking function construction. Because of the duality stated in Lemma 4.12, the ranking function construction part can be interpreted as the *dual* of algorithm ZCYCLE. Algorithm 1 makes use of this duality to achieve completeness: it either returns a ranking function, which witnesses termination, or it returns a non-negative cycle, which witnesses non-termination. The duality also means that ranking function construction comes essentially for free, as primal-dual LP solvers simultaneously generate solutions for both problems. An additional result is the improved analysis of the recursion depth: [24] uses the fact that the number of locations  $|Q|$  is a trivial upper bound of the recursion depth, while we have shown the bound  $\dim(\mathcal{A})$  (see Theorem 4.3). With this result and with LP (Q), which simultaneously solves all constraint systems  $(A_t)/(B_t)$  and thus avoids an iteration over  $t$ , we affirmatively answer the open question of Kosaraju and Sullivan [24], whether the complexity can be expressed as a polynomial function in the dimension  $d$  times the complexity of a linear program.

**Proof of Theorem 4.5.** First, we will prove the  $O$ -bound by induction on the depth of recursion (of Decompose). More precisely, if the algorithm returns *tight = true* and the depth of recursion (number of calls) is  $i$ , the termination complexity is in  $O(n^{i+1})$ .

- If there is no recursive call of procedure Decompose then QRF  $f$  obtained on line 7 is actually a RF, because  $T_f = \emptyset$  i.e., all transitions are  $f$ -ranked. Due to Theorem 4.1 we have  $\mathcal{L}_{\mathcal{A}} \in O(n)$ .
- Let  $i > 0$  be the recursion depth. Assume the claim is correct for every run of the algorithm with recursion depth  $< i$ . By induction hypothesis we have that every SCC  $\mathcal{A}_j$  of  $\mathcal{A}_{T_f}$  has termination complexity  $\mathcal{L}_{\mathcal{A}_j} \in O(n^i)$ .

Let  $q_0 \mathbf{u}_0$  be an initial configuration. Now assume we have a VASS  $\mathcal{A}$  and QRF  $f$ . If a transition is  $f$ -ranked, the  $f$ -value of the next configuration decreases by at least 1. If it is  $f$ -neutral, it does not increase. Notice that every configuration  $p\mathbf{v}$  satisfies  $f(p\mathbf{v}) = \mathbf{c}_f^\top \cdot \mathbf{v} + \mathbf{w}_f(p) \geq \mathbf{w}_f(p) \geq \min_{q \in Q} \mathbf{w}_f(q)$  since  $\mathbf{v}$  and  $\mathbf{c}_f$  are non-negative. Therefore, any zero-avoiding path can have at most  $f(q_0 \mathbf{u}_0) - \min_{q \in Q} \mathbf{w}_f(q)$  of  $f$ -ranked transitions.

Let  $g$  be the positive QRF whose existence is ensured on line 1 of the algorithm (since algorithm returns  $(k, \text{true})$ ). We give a linear bound on the size of counters in every configuration



$pv$  reachable from  $q_0u_0$ . Since  $g$  is a QRF, we have

$$g(q_0u_0) \geq g(pv),$$

$$w_g(q_0) + \sum_{i=1}^d c_g(i) \cdot u_0(i) \geq w_g(p) + \sum_{i=1}^d c_g(i) \cdot v(i).$$

Let  $c_{max} = \max_{i=1, \dots, d} c_g(i)$  and  $c_{min} = \min_{i=1, \dots, d} c_g(i)$ . Now for any counter  $j \in \{1, \dots, d\}$  we have the following estimates.

$$w_g(q_0) - w_g(p) + \sum_{i=1}^d c_{max} \cdot u_0(i) \geq c_{min} v(j),$$

$$\frac{w_g(q_0) - \min_{q \in Q} w_g(q)}{c_{min}} + d \cdot \frac{c_{max}}{c_{min}} \cdot \max_{i=1, \dots, d} u_0(i) \geq v(j).$$

Therefore, the size of any reachable configuration is linearly bounded by the size of the initial configuration and after  $O(n^i)$  transitions in some SCC we have to do at least one  $f$ -ranked transition. From this we obtain  $\mathcal{L}(\mathcal{A}) = O(n^{i+1})$  and the proof is done.

Now we want to prove that the VASS terminates even if *tight* = *false*. Again, we do the proof by induction on the depth of recursion of *Decompose*. The base of the induction is the same as in the proof of  $O$ -bound. In the induction step we assume only that every SCC of  $\mathcal{A}_{T_f}$  is terminating. Again, no transition increases the  $f$ -value and we can do only  $f(q_0u_0) - \min_{q \in Q} w_f(q)$  of  $f$ -ranked transitions, therefore  $\mathcal{A}$  is terminating (we cannot stay in one SCC indefinitely and by switching between them, we have to make at least one  $f$ -ranked transition).

**Proof of Theorem 4.6.** We now prove the correctness of our algorithm w.r.t. lower bounds. To do this, we show how to construct, for each sufficiently large  $n$ , a path of length  $\Omega(n^k)$  which results into a computation. We start with a lemma which shows a useful property of a QRF  $f$  that maximizes the number of  $f$ -ranked transitions: whenever we have a cycle consisting solely of  $f$ -neutral transitions, the effect of this cycle can be in some sense compensated by executing a combination of some other cycles.

**Lemma 4.13.** *Let  $\mathcal{A}$  be a connected VASS, and let  $f$  be a QRF for  $\mathcal{A}$  which maximizes the number of  $f$ -ranked transitions. Let  $c_f$  be the normal of  $f$ . Then for each vector  $v \in \text{Inc}$  with  $c_f^\top \cdot v = 0$  there exists a vector  $w \in \text{cone}(\text{Inc})$  such that  $v + w \geq 0$ .*

*Proof.* We consider some vector  $v \in \text{Inc}$  with  $c_f^\top \cdot v = 0$ . We have  $\text{eff}(C) = v$  for some simple cycle  $C$ . We consider the edges along the cycle  $C$ . Every edge of  $C$  must be  $f$ -neutral: otherwise we could add the equations  $c_f^\top u + w_f(p') \leq w_f(p)$  for every transition  $(p, u, p')$  along  $C$  in order to witness  $c_f^\top \cdot v < 0$ . Hence, for every transition  $t$  along  $C$  there is no other QRF  $f'$  such that  $t$  is  $f'$ -ranked; otherwise,  $f$  would not be maximal with regard to the number of  $f$ -ranked transitions. Thus, constraint system  $B_t$  is unsatisfiable for every transition  $t$  of  $C$ . By Lemma 4.12, constraint system  $A_t$  is satisfiable. For every transition  $t$  of  $C$ , we fix some non-negative multi-cycle  $M_t$  associated to some integer solution of  $A_t$ . We take the union of the non-negative multi-cycles  $M_t$  in order to obtain the non-negative multi-cycle  $M$ . We note that  $M$  contains every transition  $t$  of  $C$ . Hence,  $M$  can be decomposed into the simple cycle  $C$  and a set of simple cycles, whose effect corresponds to some vector  $w \in \text{cone}(\text{Inc})$ . Because  $M$  is non-negative, we get  $v + w \geq 0$ .  $\square$

We now proceed with the proof of Theorem 4.6. We show that if  $\mathcal{A}$  is a strongly connected VASS, and the call *Decompose*( $\mathcal{A}$ ) returns a number  $k \in \mathbb{N}$ , then for all sufficiently large  $n \in \mathbb{N}$  there exists a configuration  $p_n n$  and a computation  $\beta_n$  of length at least  $b \cdot n^k$  initiated in  $p_n n$ , where  $b \in \mathbb{Q}^+$  is a fixed positive constant independent of  $n$ . We proceed by induction on  $k$ . If  $k = 1$ , then  $\mathcal{A}$  admits a RF and the existence of such a zero-avoiding computation of linear length follows from Theorems 4.1 and 3.1. Now assume that  $k > 1$ . Then the call *Decompose*( $\mathcal{A}$ ) must result in a recursive sub-call *Decompose*( $\mathcal{A}'$ ) which returns  $k - 1$ . We prove that for all sufficiently large  $n$  there exists a computation initiated in some  $p_n n$  of length  $\Omega(n^k)$ . We prove the existence of such a path in several sub-steps.

*Constructing the paths of length  $\Omega(n^{k-1})$ .* Since the termination complexity of  $\mathcal{A}'$  is  $\Omega(n^{k-1})$  (by induction hypothesis), there is  $b \in \mathbb{R}^+$  such that for all sufficiently large  $m \in \mathbb{N}$  there exist a configuration  $p_m m$  and a computation  $\beta_m$  of length at least  $b \cdot m^{k-1}$  initiated in  $p_m m$ . Since  $\pi_{\beta_m}$  inevitably contains a cycle whose length is at least  $b' \cdot m^{k-1}$  (for some fixed  $b' \in \mathbb{R}^+$  independent of  $\beta_m$ ), we can safely assume that  $\pi_{\beta_m}$  is actually a cycle, which implies  $\text{eff}(\pi_{\beta_m}) \in \text{cone}(\text{Inc})$ .

*Constructing the compensating path.* Since  $\pi_{\beta_m}$  is such that  $c_f^\top \cdot \text{eff}(\pi_{\beta_m}) = 0$  and  $\text{eff}(\pi_{\beta_m}) \in \text{cone}(\text{Inc})$ , it follows from Lemma 4.13 that there exists  $u \in \text{cone}(\text{Inc})$  such that  $u + \text{eff}(\pi_{\beta_m}) \geq 0$ , i.e.,  $u \geq -\text{eff}(\pi_{\beta_m})$ . Since  $u = \sum_{j=1}^k a_j \cdot v_j$ , where  $k \in \mathbb{N}$ ,  $a_j \in \mathbb{Q}^+$ , and  $v_j \in \text{Inc}$  for all  $1 \leq j \leq k$ , a straightforward idea is to define the compensating path by “concatenating”  $[a_j]$  copies of  $\gamma_j$ , where  $\text{eff}(\gamma_j) = v_j$ , for all  $1 \leq j \leq k$ . This would produce the desired effect on the counters, but there is no bound on the counter decrease in intermediate configurations visited when executing this path. To overcome this problem, we choose  $m$  and construct the compensating path for  $\pi_{\beta_m}$  more carefully. We use the following lemma.

**Lemma 4.14.** *Let  $\mathcal{A}$  be a VASS and  $f$  a QRF maximizing the number of  $f$ -ranked transitions. Then there exists  $\delta \in \mathbb{R}^+$  such that for every  $m \in \mathbb{N}$  and every cycle  $\pi_m$  with  $c_f^\top \cdot \text{eff}(\pi_m) = 0$  and  $\text{eff}(\pi_m) \geq -m$  there is a path  $\varrho_m$  such that  $\text{eff}(\pi_m) + \text{eff}(\varrho_m) \geq -(d+1) \cdot |Q| \cdot \max_{\mathcal{A}}$  and no counter is decreased by more than  $\delta \cdot m$  along  $\varrho_m$ .*

*Constructing a computation  $\alpha_n$  of length  $\Omega(n^k)$ .* Now we are ready to put the above ingredients together, which still requires some effort.

Assume  $pv$  is an initial configuration. Now we need only  $|Q|$  transitions in order to get to the SCC  $\mathcal{A}'$  where we execute a path  $\pi_{\beta_m}$  with  $c_f^\top \cdot \text{eff}(\pi_{\beta_m}) = 0$  of length  $\Omega(n^{k-1})$ .

We need to choose  $m$  as large as possible but small enough so that we can execute path  $\pi_{\beta_m}$  and its compensating path  $\varrho_m$ . At the end of  $\pi_{\beta_m}$  some counters may be decreased by  $m$  to  $n - (|Q| \cdot \max_{\mathcal{A}}) - m$  (remember, we used at most  $|Q|$  transitions to get to the starting state of  $\pi_{\beta_m}$ ). Then we need to execute the compensating path  $\varrho_m$ . For this we need counters of size at most  $\delta \cdot m + |Q| \cdot \max_{\mathcal{A}}$  (we need to reach the initial state of  $\varrho_m$  and then execute this path). Together we need

$$n - |Q| \cdot \max_{\mathcal{A}} - m \geq \delta \cdot m + |Q| \cdot \max_{\mathcal{A}}.$$

We want to maximize  $m$  (in order to get a long path). This yields

$$m = \left\lfloor \frac{n - 2|Q| \cdot \max \mathcal{A}}{1 + \delta} \right\rfloor.$$

After this, every counter decreased by at most  $(d + 3)|Q| \cdot \max \mathcal{A}$  (we needed to get to the right SCC and then run the compensating path).

Repeating this procedure  $O(n)$  times, we obtain a path of length  $\Omega(n \cdot n^{k-1}) = \Omega(n^k)$ . This finishes the proof of Theorem 4.6.

**Non-strongly connected VASS.** We remark that our complete complexity classification of type (B) VASS extends to non-strongly connected VASS whose each SCC is also of type (B).

**Lemma 4.15.** *Let  $\mathcal{A}$  be a VASS,  $\mathcal{A}_1, \dots, \mathcal{A}_l$  its SCCs (reachable from the initial configuration).*

1.  $\mathcal{L}_{\mathcal{A}} \in \Omega(\max_{i \in \{1, \dots, l\}} \mathcal{L}_{\mathcal{A}_i})$
2. *If for every SCC of  $\mathcal{A}$  there is a positive QRF then  $\mathcal{L}_{\mathcal{A}} \in \Omega(\max_{i \in \{1, \dots, l\}} \mathcal{L}_{\mathcal{A}_i})$ .*

*Proof.* The first part of the lemma is trivial. Since we can visit any SCC  $\mathcal{A}_i$  in a number of transitions bounded by  $|Q|$  from the initial one, the asymptotic complexity cannot be lower than that of  $\mathcal{A}_i$ .

As in the proof of Theorem 4.5 we have for any SCC  $\mathcal{A}_i$  and any configuration  $qu$  with  $q \in Q_{\mathcal{A}_i}$  that the size of any  $pv$  with  $p \in Q_{\mathcal{A}_i}$  reachable from  $qu$  is linearly bounded by some constant depending only on the positive QRF for  $\mathcal{A}_i$ .

Since the number of SCCs for a given VASS is a constant, the size of the counter vector can increase during any computation at most by a factor independent of the size of the initial configuration. Therefore, the second claim holds.  $\square$

## 5 Conclusions

Our results open a number of interesting directions for future work. First, whether our precise complexity analysis or the complete method can be extended to other models (such as affine programs with loops) is an interesting theoretical direction to pursue. Second, our result can be used for developing a scalable tool for sound and complete analysis of asymptotic bounds for VASS.

## Acknowledgements

Tomáš Brázdil, Antonín Kučera, and Dominik Velan are supported by the Czech Science Foundation, grant No. 18-11193S. Krishnendu Chatterjee is supported by the Vienna Science and Technology Fund (WWTF) grant ICT15-003; and an ERC Start Grant (279307: Graph Games). Krishnendu Chatterjee and Florian Zuleger are supported by the Austrian Science Fund (FWF) grant: S11407-N23 (RiSE).

## References

- [1] E. Albert, P. Arenas, S. Genaim, M. Gómez-Zamalloa, G. Puebla, D. V. Ramírez-Deantes, G. Román-Díez, and D. Zanardini. 2009. Termination and Cost Analysis with COSTA and its User Interfaces. *ENTCS* 258, 1 (2009), 109–121.
- [2] C. Alias, A. Darte, P. Feautrier, and L. Gonnord. 2010. Multi-dimensional Rankings, Program Termination, and Complexity Bounds of Flowchart Programs. In *Proc. of SAS 2010*. 117–133.
- [3] B. Aminof, S. Rubin, and F. Zuleger. 2015. On the Expressive Power of Communication Primitives in Parameterised Systems. In *Proc. of LPAR 2015*. 313–328.
- [4] B. Aminof, S. Rubin, F. Zuleger, and F. Spegini. 2015. Liveness of Parameterized Timed Networks. In *Proc. of ICALP 2015*. 375–387.
- [5] M. F. Atig and P. Habermehl. 2011. On Yen's path logic for Petri nets. *International Journal of Foundations of Computer Science* 22, 04 (2011), 783–799.
- [6] R. Bloem, S. Jacobs, A. Khalimov, I. Konnov, S. Rubin, H. Veith, and J. Widder. 2016. Decidability in Parameterized Verification. *SIGACT News* 47, 2 (2016), 53–64.
- [7] L. Bozzelli and P. Ganty. 2011. Complexity Analysis of the Backward Coverability Algorithm for VASS. In *Proc. of RP 2011*. 96–109. Tomáš Brázdil, Krishnendu Chatterjee, Antonín Kučera, Petr Novotný, Dominik Velan, Florian Zuleger
- [8] T. Brázdil, C. Chatterjee, A. Kučera, P. Novotný, D. Velan, and F. Zuleger. 2018. Efficient Algorithms for Asymptotic Bounds on Termination Time in VASS. *arXiv:1804.10985*.
- [9] Q. Carbonneaux, J. Hoffmann, T. W. Reps, and Z. Shao. 2017. Automated Resource Analysis with Coq Proof Objects. In *Proc. of CAV 2017*. 64–85.
- [10] K. Chatterjee, L. Doyen, T. A. Henzinger, and J-F. Raskin. 2010. Generalized Mean-payoff and Energy Games. In *Proc. of FSTTCS 2010*. 505–516.
- [11] K. Chatterjee and Y. Velner. 2013. Hyperplane Separation Technique for Multi-dimensional Mean-Payoff Games. In *Proc. of CONCUR 2013*. 500–515.
- [12] T. Colcombet, M. Jurdzinski, R. Lazic, and S. Schmitz. 2017. Perfect Half Space Games. In *Proc. of LICS 2017*. 1–11.
- [13] J. Esparza. 1998. Decidability and complexity of Petri net problems – An introduction. *Lectures on Petri nets I: Basic models* (1998), 374–428.
- [14] J. Esparza, R. Ledesma-Garza, R. Majumdar, P. Meyer, and F. Niksic. 2014. An SMT-Based Approach to Coverability Analysis. In *Proc. of CAV 2014*. 603–619.
- [15] J. Esparza and M. Nielsen. 1994. Decidability Issues for Petri Nets – A survey. *Bulletin of the EATCS* 52 (1994), 245–262.
- [16] A. F. Montoya and R. Hähnle. 2014. Resource Analysis of Complex Programs with Cost Equations. In *Proc. of APLAS 2014*. 275–295.
- [17] J. Giesl, C. Aschermann, M. Brockschmidt, F. Emmes, F. Frohn, C. Fuhs, J. Hensel, C. Otto, M. Plücker, P. Schneider-Kamp, T. Ströder, S. Swiderski, and R. Thiemann. 2017. Analyzing Program Termination and Complexity Automatically with AProVE. *J. Autom. Reasoning* 58, 1 (2017), 3–31.
- [18] S. Gulwani, K. K. Mehra, and T. M. Chilikimbi. 2009. SPEED: precise and efficient static estimation of program computational complexity. In *Proc. of POPL 2009*. 127–139.
- [19] S. Gulwani and F. Zuleger. 2010. The reachability-bound problem. In *Proc. of PLDI 2010*. 292–304.
- [20] J. Hoffmann, K. Aehlig, and M. Hofmann. 2012. Multivariate amortized resource analysis. *ACM Trans. Program. Lang. Syst.* 34, 3 (2012), 14:1–14:62.
- [21] M. Jurdzinski, R. Lazic, and S. Schmitz. 2015. Fixed-Dimensional Energy Games are in Pseudo-Polynomial Time. In *Proc. of ICALP 2015*. 260–272.
- [22] R. M. Karp and R. E. Miller. 1969. Parallel Program Schemata. *J. Comput. Syst. Sci.* 3, 2 (1969), 147–195.
- [23] S. R. Kosaraju. 1982. Decidability of Reachability in Vector Addition Systems (Preliminary Version). In *Proc. of STOC 1982*. ACM, 267–281.
- [24] S. R. Kosaraju and G. F. Sullivan. 1988. Detecting Cycles in Dynamic Graphs in Polynomial Time. In *Proc. of STOC 1988*. 398–406.
- [25] J. Leroux. 2011. Vector Addition System Reachability Problem: A Short Self-contained Proof. In *Proc. of POPL 2011*. ACM, 307–316.
- [26] J. Leroux and S. Schmitz. 2015. Demystifying Reachability in Vector Addition Systems. In *Proc. of LICS 2015*. 56–67.
- [27] J. Leroux and P. Schnoebelen. 2014. On Functions Weakly Computable by Petri Nets and Vector Addition Systems. In *Proc. of RP 2014*. 190–202.
- [28] R. Lipton. 1976. *The Reachability Problem Requires Exponential Space*. Yale Technical report 62.
- [29] E. Mayr. 1984. An Algorithm for the General Petri Net Reachability Problem. *SIAM-JC* 13 (1984), 441–460.
- [30] A. Podelski and A. Rybalchenko. 2004. A Complete Method for the Synthesis of Linear Ranking Functions. In *Proc. of VMCAI 2004*. 239–251.
- [31] C. Rackoff. 1978. The Covering and Boundedness Problems for Vector Addition Systems. *Theor. Comp. Sci.* 6 (1978), 223–231.
- [32] S. Schmitz. 2016. Complexity Hierarchies Beyond Elementary. *ACM Trans. Comput. Theory* 8, 1 (Feb. 2016), 3:1–3:36.
- [33] M. Sinn, F. Zuleger, and H. Veith. 2014. A Simple and Scalable Static Analysis for Bound Analysis and Amortized Complexity Analysis. In *Proceedings of CAV 2014*. 745–761.
- [34] M. Sinn, F. Zuleger, and H. Veith. 2017. Complexity and Resource Bound Analysis of Imperative Programs Using Difference Constraints. *J. Autom. Reasoning* 59, 1 (2017), 3–45.
- [35] Y. Velner, K. Chatterjee, L. Doyen, T. A. Henzinger, A. M. Rabinovich, and J-F. Raskin. 2015. The complexity of multi-mean-payoff and multi-energy games. *Inf. Comput.* 241 (2015), 177–196.
- [36] L. Yang, C. Zhou, N. Zhan, and B. Xia. 2010. Recent advances in program verification through computer algebra. *Frontiers of Computer Science in China* 4, 1 (2010), 1–16.
- [37] H.-C. Yen. 1992. A unified approach for deciding the existence of certain Petri net paths. *Inf. Comput.* 96, 1 (1992), 119–137.