Long-Run Average Behavior of Vector Addition Systems with States

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

Thomas A. Henzinger
IST Austria, Klosterneuburg, Austria
tah@ist.ac.at

Jan Otop
University of Wrocław, Poland
jotop@cs.uni.wroc.pl

Abstract

A vector addition system with states (VASS) consists of a finite set of states and counters. A configuration is a state and a value for each counter; a transition changes the state and each counter is incremented, decremented, or left unchanged. While qualitative properties such as state and configuration reachability have been studied for VASS, we consider the long-run average cost of infinite computations of VASS. The cost of a configuration is for each state, a linear combination of the counter values. In the special case of uniform cost functions, the linear combination is the same for all states. The (regular) long-run emptiness problem is, given a VASS, a cost function, and a threshold value, if there is a (lasso-shaped) computation such that the long-run average value of the cost function does not exceed the threshold. For uniform cost functions, we show that the regular long-run emptiness problem is (a) decidable in polynomial time for integer-valued VASS, and (b) decidable but nonelementarily hard for natural-valued VASS (i.e., nonnegative counters). For general cost functions, we show that the problem is (c) NP-complete for integer-valued VASS, and (d) undecidable for natural-valued VASS. Our most interesting result is for (c) integer-valued VASS with general cost functions, where we establish a connection between the regular long-run emptiness problem and quadratic Diophantine inequalities. The general (nonregular) long-run emptiness problem is equally hard as the regular problem in all cases except (c), where it remains open.

2012 ACM Subject Classification Theory of computation → Automata over infinite objects; Theory of computation → Quantitative automata

Keywords and phrases vector addition systems, mean-payoff, Diophantine inequalities

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2019.27

Related Version A full version of the paper is available at `http://arxiv.org/abs/1905.05537`.

Funding *Krishnendu Chatterjee*: The Austrian Science Fund (FWF) NFN grant S11407-N23 (RiSE/SHiNE)
Thomas A. Henzinger: The Austrian Science Fund (FWF) grants S11402-N23 (RiSE/ShiNE) and Z211-N23 (Wittgenstein Award)
Jan Otop: The National Science Centre (NCN), Poland under grant 2017/27/B/ST6/00299

Acknowledgements We thank Petr Novotný for helpful discussion on the literature.

1 Introduction

Vector Addition System with States (VASS). *Vector Addition Systems (VASS)* [20] are equivalent to Petri Nets, and they provide a fundamental framework for formal analysis of parallel processes [14]. The extension of VASs with a finite-state transition structure

30th International Conference on Concurrency Theory (CONCUR 2019).
Editors: Wan Fokkink and Rob van Glabbeek; Article No. 27; pp. 27:1–27:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

gives *Vector Addition Systems with States (VASS).* Informally, a VASS consists of a finite set of control states and transitions between them, and a set of k counters, where at every transition between the control states each counter is either incremented, decremented, or remains unchanged. A *configuration* consists of a control state and a valuation of each counter, and the transitions of the VASS describe the transitions between the configurations. Thus a VASS represents a finite description of an infinite-state transition system between the configurations. If the counters can hold all possible integer values, then we call them integer-valued VASS; and if the counters can hold only non-negative values, then we call them natural-valued VASS.

Modelling power. VASS are a fundamental model for concurrent processes [14], and often used in performance analysis of concurrent processes [12, 16, 18, 19]. Moreover, VASS have been used (a) in analysis of parametrized systems [2], (b) as abstract models for programs for bounds and amortized analysis [29], (c) in interactions between components of an API in component-based synthesis [15]. Thus they provide a rich framework for a variety of problems in verification as well as program analysis.

Previous results for VASS. A computation (or a run) in a VASS is a sequence of configurations. The well-studied problems for VASS are as follows: (a) *control-state reachability* where given a set of target control states a computation is successful if one of the target states is reached; (b) *configuration reachability* where given a set of target configurations a computation is successful if one of the target configurations is reached. For natural-valued VASS, (a) the control-state reachability problem is ExpSpace-complete: the ExpSpace-hardness is shown in [25, 13] and the upper bound follows from [28]; and (b) the configuration reachability problem is decidable [26, 21, 22, 23], and a recent breakthrough result shows that the problem is non-elementary hard [10]. For integer-valued VASS, (a) the control-state reachability problem is NLogSpace-complete: the counters can be abstracted away and we have to solve the reachability problem for graphs; and (b) the configuration reachability problem is NP-complete: this is a folklore result obtained via reduction to linear Diophantine inequalities.

Long-run average property. The classical problems for VASS are qualitative (or Boolean) properties where each computation is either successful or not. In this work we consider long-run average property that assigns a real value to each computation. A cost is associated to each configuration and the value of a computation is the long-run average of the costs of the configurations of the computation. For cost assignment to configuration we consider linear combination with natural coefficients of the values of the counters. In general the linear combination for the cost depends on the control state of the configuration, and in the special case of uniform cost functions the linear combination is the same across all states.

Motivating examples. We present some motivating examples for the problems we consider. First, consider a VASS where the counters represent different queue lengths, and each queue consumes resource (e.g., energy) proportional to its length, however, for different queues the constant of the proportionality might differ. The computation of the long-run average resource consumption is modeled as a uniform cost function, where the linear combination for the cost function is obtained from the constants of proportionality. Second, consider a system that uses two different batteries, and the counters represent the charge levels. At different states, different batteries are used, and we are interested in the long-run average charge of the used battery. This is modeled as a general cost function, where depending on the control state the cost function is the value of the counter representing the battery used in the state.

Our contributions. We consider the following decision problem: given a VASS and a cost function decide whether there is a regular (or periodic) computation such that the long-run average value is at most a given threshold. Our main contributions are as follows:

1. For uniform cost functions, we show that the problem is (a) decidable in polynomial time for integer-valued VASS, and (b) decidable but non-elementary hard for natural-valued VASS. In (b) we assume that the cost function depends on all counters.
2. For general cost functions, we show that the problem is (a) NP-complete for integer-valued VASS, and (b) undecidable for natural-valued VASS.

Our most interesting result is for general cost functions and integer-valued VASS, where we establish an interesting connection between the problem we consider and quadratic Diophantine inequalities. Finally, instead of regular computations, if we consider existence of an arbitrary computation, then all the above results hold, other than the NP-completeness result for general cost functions and integer-valued VASS (which remains open).

Related works. Long-run average behavior have been considered for probabilistic VASS [5], and other infinite-state models such as pushdown automata and games [9, 8, 1]. In these works the costs are associated with transitions of the underlying finite-state representation. In contrast, in our work the costs depend on the counter values and thus on the configurations. Costs based on configurations, specifically the content of the stack in pushdown automata, have been considered in [27]. Quantitative asymptotic bounds for polynomial-time termination in VASS have also been studied [4, 24], however, these works do not consider long-run average property. Finally, a related model of automata with monitor counters with long-run average property have been considered in [7, 6]. However, there are some crucial differences: in automata with monitor counters, the cost always depends on one counter, and counters are reset once the value is used. Moreover, the complexity results for automata with monitor counters are quite different from the results we establish.

2 Preliminaries

For a sequence w, we define $w[i]$ as the $(i+1)$-th element of w (we start with 0) and $w[i,j]$ as the subsequence $w[i]w[i+1]\dots w[j]$. We allow j to be ∞ for infinite sequences. For a finite sequence w, we denote by $|w|$ its length; and for an infinite sequence the length is ∞.

We use the same notation for vectors. For a vector $\vec{x} \in \mathbb{R}^k$ (resp., $\mathbb{Q}^k$, $\mathbb{Z}^k$ or $\mathbb{N}^k$), we define $x[i]$ as the i-th component of $\vec{x}$. We define the support of $\vec{x}$, denoted by $supp(\vec{x})$ as the set of components of $\vec{x}$ with non-zero values. For vectors $\vec{x}, \vec{y}$ of equal dimension, we denote by $\vec{x} \cdot \vec{y}$, the dot-product of $\vec{x}$ and $\vec{y}$.

2.1 Vector addition systems with states (VASS)

A k-dimensional vector addition system with states (VASS) over $\mathbb{Z}$ (resp., over $\mathbb{N}$), referred to as VASS($\mathbb{Z}, k$) (resp., VASS($\mathbb{N}, k$)), is a tuple $\mathcal{A} = \langle Q, Q_0, \delta \rangle$, where (1) Q is a finite set of states, (2) $Q_0 \subseteq Q$ is a set of initial states, and (3) $\delta \subseteq Q \times Q \times \mathbb{Z}^k$. We denote by $\mathbf{VASS}(\mathbb{Z}, k)$ (resp., $\mathbf{VASS}(\mathbb{N}, k)$) the class of k-dimensional VASS over $\mathbb{Z}$ (resp., $\mathbb{N}$). We often omit the dimension in VASS and write VASS($\mathbb{Z}$), VASS($\mathbb{N}$), $\mathbf{VASS}(\mathbb{N})$, $\mathbf{VASS}(\mathbb{Z})$ if a definition or an argument is uniform w.r.t. the dimension.

Configurations and computations. A *configuration* of a $\mathrm{VASS}(\mathbb{Z}, k)$ $\mathcal{A}$ is a pair from $Q \times \mathbb{Z}^k$, which consists of a state and a valuation of the counters. A *computation* of $\mathcal{A}$ is an infinite sequence π of configurations such that (a) $\pi[0] \in Q_0 \times \{\vec{0}\}$[1], and (b) for every $i \geq 0$, there exists $(q, q', \vec{y}) \in \delta$ such that $\pi[i] = (q, \vec{x})$ and $\pi[i+1] = (q', \vec{x} + \vec{y})$. A computation of a $\mathrm{VASS}(\mathbb{N}, k)$ $\mathcal{A}$ is a computation π of $\mathcal{A}$ considered as a $\mathrm{VASS}(\mathbb{Z}, k)$ such that the values of all counters are natural, i.e., for all $i > 0$ we have $\pi[i] \in Q \times \mathbb{N}^k$.

Paths and cycles. A path $\rho = (q_0, q_0', \vec{y}_0), (q_1, q_1', \vec{y}_1), \dots$ in a $\mathrm{VASS}(\mathbb{Z})$ (resp., $\mathrm{VASS}(\mathbb{N})$) $\mathcal{A}$ is a (finite or infinite) sequence of transitions (from δ) such that for all $0 \leq i \leq |\rho|$ we have $q_i' = q_{i+1}$. A finite path ρ is a cycle if $\rho = (q_0, q_0', \vec{y}_0), \dots, (q_m, q_m', \vec{y}_m)$ and $q_0 = q_m'$. Every computation in a $\mathrm{VASS}(\mathbb{Z})$ (resp., $\mathrm{VASS}(\mathbb{N})$) defines a unique infinite path. Conversely, every infinite path in a $\mathrm{VASS}(\mathbb{Z})$ $\mathcal{A}$ starting with $q_0 \in Q_0$ defines a computation in $\mathcal{A}$. However, if $\mathcal{A}$ is a $\mathrm{VASS}(\mathbb{N}, k)$, some paths do not have corresponding computations due to non-negativity restriction posed on the counters.

Regular computations. We say that a computation π of a $\mathrm{VASS}(\mathbb{Z})$ (resp., $\mathrm{VASS}(\mathbb{N})$) is *regular* if it corresponds to a path which is ultimately periodic, i.e., it is of the form $\alpha\beta^{\omega}$ where α, β are finite paths.

2.2 Decision problems

We present the decision problems that we study in the paper.

Cost functions. Consider a $\mathrm{VASS}(\mathbb{Z}, k)$ (resp., $\mathrm{VASS}(\mathbb{N}, k)$) $\mathcal{A} = (Q, Q_0, \delta)$. A cost function f for $\mathcal{A}$ is a function $f \colon Q \times \mathbb{Z}^k \to \mathbb{Z}$, which is linear with natural coefficients, i.e., for every q there exists $\vec{a} \in \mathbb{N}^k$ such that $f(q, \vec{z}) = \vec{a} \cdot \vec{z}$ for all $\vec{z} \in \mathbb{Z}^k$. We extend cost functions to computations as follows. For a computation π of $\mathcal{A}$, we define $f(\pi)$ as the sequence $f(\pi[0]), f(\pi[1]), \dots$. Every cost function f is given by a labeling $l \colon Q \to \mathbb{N}^k$ by $f(q, \vec{z}) = l(q) \cdot \vec{z}$. We define the size of f, denoted by $|f|$, as the size of binary representation of l considered as a sequence of natural numbers of the length $|Q| \cdot k$.

Uniform cost functions. We say that a cost function f is *uniform*, if it is given by a constant function l, i.e., for all states q, q' and $\vec{z} \in \mathbb{Z}^k$ we have $f(q, \vec{z}) = f(q', \vec{z})$. Uniform cost functions are given by a single vector $\vec{a} \in \mathbb{N}^k$.

The long-run average. We are interested in the long-run average of the values returned by the cost function, which is formalized as follows. Consider an infinite sequence of real numbers w. We define $\textsc{LimAvg}(w) = \liminf_{k \to \infty} \frac{1}{k+1} \sum_{i=0}^{k} w[i]$.

▶ **Definition 1** (The average-value problems). *Given a* $\mathrm{VASS}(\mathbb{Z}, k)$ *(resp.,* $\mathrm{VASS}(\mathbb{N}, k)$*)* $\mathcal{A}$*, a cost function* f *for* $\mathcal{A}$*, and a threshold* $\lambda \in \mathbb{Q}$*,*

- *the **average-value** problem (resp., the **regular average-value** problem) asks whether* $\mathcal{A}$ *has a computation (resp., a regular computation)* π *such that* $\textsc{LimAvg}(f(\pi)) \leq \lambda$*, and*
- *the **finite-value** problem (resp., the **regular finite-value** problem) asks whether* $\mathcal{A}$ *has a computation (resp., a regular computation)* π *such that* $\textsc{LimAvg}(f(\pi)) < \infty$*.*

The following example illustrates the regular average-value problem for $\mathbf{VASS}(\mathbb{Z})$.

[1] Without loss of generality, we assume that the initial counter valuation is $\vec{0}$. We can encode any initial configuration in the VASS itself.

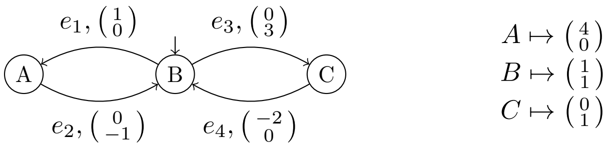

Figure 1 The VASS $\mathcal{A}_e$ and its labeling.

▶ **Example 2.** Consider the VASS$(\mathbb{Z}, 2)$ $\mathcal{A}_e = \langle \{A, B, C\}, \{B\}, \{e_1, e_2, e_3, e_4\}\rangle$ depicted in Figure 1 and a (non-uniform) cost function f is given by the labeling from Figure 1. $A \mapsto \binom{4}{0}$,

Consider an infinite path $(e_1e_2e_3e_4)^\omega$ that defines the regular computation π_0. This computation can be divided into blocks of 4 consecutive configurations. The $(i+1)$-th block has the following form

$$\ldots(B, \binom{-i}{2i})(A, \binom{-i+1}{2i})(B, \binom{-i+1}{2i-1})(C, \binom{-i+1}{2i+2})\ldots$$

and the corresponding values of $f(\pi)$ are:

$$\ldots \quad i \quad -4i+4 \qquad i \qquad 2i+2 \quad \ldots$$

Therefore, the sum of values over each block is 6 and hence $\textsc{LimAvg}(f(\pi_0)) = \frac{3}{2}$.

The answer to the (regular) average-value problem with any threshold is YES. Consider $j \in \mathbb{N}$ and a path $(e_1e_2)^j(e_1e_2e_3e_4)^\omega$. Observe that it defines a regular computation π_j, which after the block $(e_1e_2)^j$ coincides with π_0 with all counter values shifted by $\binom{-2j}{3j}$. Note that in each block $e_1e_2e_3e_4$, the value f on the vector $\binom{-2j}{3j}$ are $j, -8j, j, 3j$. Therefore, $\textsc{LimAvg}(f(\pi_j)) = \textsc{LimAvg}(f(\pi_0)) + \frac{-3j}{4} = \frac{-3j-6}{4}$. It follows that for every threshold $\lambda \in \mathbb{Q}$, there exists a regular computation π_j with $\textsc{LimAvg}(f(\pi_j)) \leq \lambda$.

Organization. In this paper, we study the (regular) average-value problem for $\mathbf{VASS}(\mathbb{Z}, k)$ and $\mathbf{VASS}(\mathbb{N}, k)$. First, we study the average-value problem for uniform cost functions (Section 3). Next, we consider the non-uniform case, where we focus on $\mathbf{VASS}(\mathbb{Z})$ (Section 4). We start with solving the (regular) finite-value problem, and then we move to the regular average-value problem. We show that both problems are NP-complete. Next, we discuss the non-uniform case for $\mathbf{VASS}(\mathbb{N})$ and we show that the regular finite-value problem is decidable (and non-elementary), while the (regular) average-value problem is undecidable.

3 Uniform cost functions

In this section we study the average-value problem for $\mathbf{VASS}(\mathbb{Z})$ and $\mathbf{VASS}(\mathbb{N})$ for the uniform cost functions.

3.1 Integer-valued VASS: VASS$(\mathbb{Z})$

Consider a VASS$(\mathbb{Z}, k)$ $\mathcal{A}$ and a uniform cost function f for $\mathcal{A}$. This function is defined by a single vector of coefficients $\vec{a} \in \mathbb{N}^k$. First, observe that we can reduce the number of the counters to one, which tracks the current cost. This counter c stores the value of $f(\pi[i]) = \vec{a} \cdot \vec{z}$, where $\vec{z}$ are the values of counters. Initially, the counter c is $0 = \vec{a} \cdot \vec{0}$. Next, in each step $i > 0$, counters $\vec{z}$ are updated with some values $\vec{y}$ and we update c with $\vec{a} \cdot \vec{y}$. Note that the value of $f(\pi[i+1]) = \vec{a} \cdot (\vec{z} + \vec{y})$ equals $f(\pi[i]) + \vec{a} \cdot \vec{y}$.

Second, observe that the average-value problem for $\mathbf{VASS}(\mathbb{Z}, 1)$ and uniform cost functions is equivalent to single-player average energy games (with no bounds on energy levels), which are solvable in polynomial time [3]. Moreover, average-energy games (with no bounds on energy levels) admit memoryless winning strategies and hence the average-value and the regular average-value problems coincide. In consequence we have:

▶ **Theorem 3.** *The average-value and the regular average-value problems for* $\boldsymbol{VASS}(\mathbb{Z})$ *and uniform cost functions are decidable in polynomial time.*

3.2 Natural-valued VASS: VASS$(\mathbb{N})$

For natural-valued VASS, we cannot reduce the number of counters to one; we need to track all counters to make sure that all of them have non-negative values. Moreover, we show that the average-value problem for (single counter) $\mathbf{VASS}(\mathbb{N}, 1)$ is in PSPACE while the average-value problem for $\mathbf{VASS}(\mathbb{N})$ is nonelementary.

First, the average-value problem for (single counter) $\mathbf{VASS}(\mathbb{N}, 1)$ in the uniform case is equivalent to single-player average energy games with non-negativity constraint on the energy values. The latter problem is in PSPACE and it is NP-hard [3].

In the multi-counter case, we show that the average-value problem is mutually reducible to the configuration reachability problem for VASS, which has recently been shown nonelementary hard [10]. We additionally assume that the cost function depends on all its arguments, i.e., all coefficients are non-negative. This assumption allows us show that if there is a computation of the average value below some λ, then there is one, which is lasso-shaped and the cycle in the lasso has exponential size in the size of the VASS and λ. Therefore, we can non-deterministically pick such a cycle and check whether it is reachable from the initial configuration.

▶ **Theorem 4.** *The average-value and the regular average-value problems for* $\boldsymbol{VASS}(\mathbb{N})$ *and uniform cost functions with non-zero coefficients are decidable and mutually reducible to the configuration reachability problem for* $\boldsymbol{VASS}(\mathbb{N})$*.*

We have used in the proof the fact that f depends on all counters. We conjecture that this assumption can be lifted:

▶ **Open question 5.** *Is the average-value problem for* $\boldsymbol{VASS}(\mathbb{N})$ *and uniform cost functions decidable?*

4 General cost functions and VASS$(\mathbb{Z})$

First, we consider (an extension of) the regular finite-value problem for $\mathbf{VASS}(\mathbb{Z})$ (Section 4.1). We show that one can decide in non-deterministic polynomial time whether a given VASS has a regular computation (a) of the value $-\infty$, or (b) of some finite value. To achieve this, we introduce path summarizations, which allow us to state conditions that entail (a) and respectively (b). We show that these conditions can be checked in NP.

We apply the results from Section 4.1 to solve the regular average-value problem (Section 4.2). We consider only VASS that have some computation of a finite value, and no computation of the value $-\infty$, i.e., the answer to (a) is NO and the answer to (b) is YES. In other cases we can easily answer to the regular average-value problem. If the answer to (a) is YES, then for any threshold we answer YES. If the answers to (a) and (b) are NO, then for any threshold we answer NO. Finally, we show NP-hardness of the regular finite-value and the regular average-value problems (Section 4.3).

The main result of this section is the following theorem:

▶ **Theorem 6.** *The regular finite-value and the regular average-value problems for* $\boldsymbol{VASS}(\mathbb{Z})$ *with (general) cost functions are* NP*-complete.*

We fix a VASS$(\mathbb{Z}, k)$ $\mathcal{A} = \langle Q, Q_0, \delta \rangle$, with the set of states Q and the set of transitions δ, and a cost function f, which we refer to throughout this section. We exclude the complexity statements, where the asymptotic behavior applies to all $\mathcal{A}$ and f.

4.1 The finite-value problem

For a path ρ, we define characteristics $\textsc{Gain}(\rho), \textsc{Vals}(\rho)$, which summarize the impact of ρ on the values of counters ($\textsc{Gain}$) and the value of the (partial) average of costs ($\textsc{Vals}$).

Let ρ be of the form $(q_1, q_2, \vec{y}_1) \dots (q_m, q_{m+1}, \vec{y}_m)$. We define $\textsc{Gain}(\rho)$ as the sum of updates along ρ, i.e., $\textsc{Gain}(\rho) = \sum_{i=1}^{m} \vec{y}_i$. The vector $\textsc{Gain}(\rho)$ is the update of counters upon the whole path ρ. Observe that $\textsc{Gain}(\rho_1\rho_2) = \textsc{Gain}(\rho_1) + \textsc{Gain}(\rho_2)$.

Let $l \colon Q \to \mathbb{N}^k$ be the function representing f, i.e., for every $\vec{z} \in \mathbb{Z}^k$ we have $f(q, \vec{z}) = l(q) \cdot \vec{z}$. We define $\textsc{Vals}(\rho)$ as the sum of vectors $l(q_i)$ along ρ, i.e., $\textsc{Vals}(\rho) = \sum_{i=1}^{m} l(q_i)$. Note that we exclude the last state q_{m+1}, and hence we have $\textsc{Vals}(\rho_1\rho_2) = \textsc{Vals}(\rho_1) + \textsc{Vals}(\rho_2)$. The vector $\textsc{Vals}(\rho)$ describes the coefficients with which each counter contributes to the average along the path ρ.

Consider a regular computation π and a path $\rho_1(\rho_2)^\omega$ that corresponds to it. Let π' be the (regular) computation obtained from π by contracting $|\rho_2|$ to a single transition $(q_f, q_f, \textsc{Gain}(\rho_2))$, which is a loop over a fresh state q_f. The counters over this transition are updated by $\textsc{Gain}(\rho_2)$ and the cost function in q_f is defined as $f(q_f, \vec{z}) = \frac{1}{|\rho_2|}\textsc{Vals}(\rho_2) \cdot \vec{z}$.

The values of π and π' may be different, but they differ only by some finite value. Indeed, the difference over a single iteration of ρ_2 updates and computation of the partial averages are interleaved along ρ_2, while in $(q_f, q_f, \textsc{Gain}(\rho_2))$ we first compute the partial average and then update the counters. Therefore, that difference is a finite value N that does not depend on the initial values of counters. It follows that the difference between the average values of (the computations corresponding to) ρ_2 and $(q_f, q_f, \textsc{Gain}(\rho_2))$ is bounded by N.

Finally, observe that the value of π' can be easily estimated. Observe that the partial average of the first $n+1$ values of f in π' equals

$$\frac{1}{n+1}\Big(\vec{0} \cdot \textsc{Vals}(\rho_2) + \textsc{Gain}(\rho_2) \cdot \textsc{Vals}(\rho_2) + \ldots + n\textsc{Gain}(\rho_2) \cdot \textsc{Vals}(\rho_2)\Big) = \frac{n}{2}\textsc{Gain}(\rho_2) \cdot \textsc{Vals}(\rho_2).$$

In consequence, we have the following:

▶ **Lemma 7.** *Let π be a regular computation corresponding to a path $\rho_1(\rho_2)^\omega$. Then, one of the following holds:*

1. $\textsc{Gain}(\rho_2) \cdot \textsc{Vals}(\rho_2) < 0$ *and* $\textsc{LimAvg}(f(\pi)) = -\infty$*, or*
2. $\textsc{Gain}(\rho_2) \cdot \textsc{Vals}(\rho_2) = 0$ *and* $\textsc{LimAvg}(f(\pi))$ *is finite, or*
3. $\textsc{Gain}(\rho_2) \cdot \textsc{Vals}(\rho_2) > 0$ *and* $\textsc{LimAvg}(f(\pi)) = \infty$*.*

▶ **Example 8.** Consider the VASS $\mathcal{A}_e$ and the cost function f from Example 2. We have shown that the computation defined by the path $(e_1e_2e_3e_4)^\omega$ has finite average value. This can be algorithmically computed using Lemma 7. We compute

$$\textsc{Gain}(e_1e_2e_3e_4) = \binom{1}{0} + \binom{0}{-1} + \binom{0}{3} + \binom{-2}{0} = \binom{-1}{2}$$
$$\textsc{Vals}(e_1e_2e_3e_4) = \binom{1}{1} + \binom{4}{0} + \binom{1}{1} + \binom{0}{1} = \binom{6}{3}$$

Therefore, $\textsc{Gain}(e_1e_2e_3e_4) \cdot \textsc{Vals}(e_1e_2e_3e_4) = 0$.

Reduction to integer quadratic programming. Lemma 7 reduces the regular finite-value problem to finding a cycle with $\textsc{Gain}(\rho) \cdot \textsc{Vals}(\rho) < 0$ (resp., $\textsc{Gain}(\rho) \cdot \textsc{Vals}(\rho) = 0$). We show that existence of such a cycle can be stated as a polynomial-size instance of *integer quadratic programming*, which can be decided in NP [11]. An instance of integer quadratic programming consists of a symmetric matrix $\mathbf{A} \in \mathbb{Q}^{n\times n}, \vec{a} \in \mathbb{Q}^n, d \in \mathbb{Q}, \mathbf{B} \in \mathbb{Q}^{n\times m}, \vec{c} \in \mathbb{Q}^m$ and it asks whether there exists a vector $\vec{x} \in \mathbb{Z}^n$ satisfying the following system:

$$\begin{aligned} \vec{x}^T\mathbf{A}\vec{x} + \vec{a}\vec{x} + d &\leq 0 \\ \mathbf{B}\vec{x} &\leq \vec{c} \end{aligned}$$

Let ρ be a cycle. Observe that both $\textsc{Gain}(\rho), \textsc{Vals}(\rho)$ depend only on the multiplicity of transitions occurring in ρ; the order of transitions is irrelevant. Consider a vector $\vec{x} = (x[1], \ldots, x[m])$ of multiplicities of transitions $e_1, \ldots, e_m$ in ρ. Then,

$$\textsc{Gain}(\rho) \cdot \textsc{Vals}(\rho) = \sum_{1 \leq i,j \leq m} x[i]x[j]\textsc{Gain}(e_i) \cdot \textsc{Vals}(e_j)$$

Therefore, we have

$$2 \cdot \textsc{Gain}(\rho) \cdot \textsc{Vals}(\rho) = \vec{x}^T\mathbf{A}\vec{x} \tag{1}$$

for a symmetric matrix $\mathbf{A} \in \mathbb{Z}^{m\times m}$ defined for all i, j as

$$\mathbf{A}[i,j] = \textsc{Gain}(e_i) \cdot \textsc{Vals}(e_j) + \textsc{Gain}(e_j) \cdot \textsc{Vals}(e_i). \tag{2}$$

The left hand side of (1) is multiplied by 2 to avoid division by 2. It follows that if $\textsc{Gain}(\rho) \cdot \textsc{Vals}(\rho) < 0$ (resp., $\textsc{Gain}(\rho) \cdot \textsc{Vals}(\rho) = 0$), then the inequality $\vec{x}^T\mathbf{A}\vec{x} - 1 \leq 0$ (resp., $\vec{x}^T\mathbf{A}\vec{x} \leq 0$) has a solution.

Note that the above inequality can have a solution, which does not correspond to a cycle. We can encode with a system of linear inequalities that $\vec{x}$ corresponds to a cycle. Consider $S \subseteq Q$, which corresponds to all states visited by ρ. It suffices to ensure that (a) for all $s \in S$, the number of incoming transitions to s, which is the sum of multiplicities of transitions leading to s, equals the number of transitions outgoing from s, (b) for every $s \in S$, the number of outgoing transitions is greater or equal to 1, and (c) for $s \in Q \setminus S$, the number of incoming and outgoing transitions is 0. Finally, we state that all multiplicities are non-negative. We can encode such equations and inequalities as $\mathbf{B}_S\vec{x} \leq \vec{c}_S$. Observe that every vector of multiplicities $\vec{x} \in \mathbb{Z}$ satisfies $\vec{x}^T\mathbf{A}\vec{x} - 1 \leq 0$ (resp., $\vec{x}^T\mathbf{A}\vec{x} \leq 0$) and $\mathbf{B}_S\vec{x} \leq \vec{c}_S$ defines a cycle ρ with $\textsc{Gain}(\rho), \textsc{Vals}(\rho) < 0$ (resp., $\textsc{Gain}(\rho), \textsc{Vals}(\rho) = 0$). The matrices $\mathbf{A}, \mathbf{B}_S$ and the vector $\vec{c}_S$ are polynomial in $|\mathcal{A}|$. The set S can be picked non-deterministically. In consequence, we have the following:

▶ **Lemma 9.** *The problem: given a* VASS$(\mathbb{Z}, k)$ $\mathcal{A}$ *and a cost function* f*, decide whether* $\mathcal{A}$ *has a regular computation of the value* $-\infty$ *(resp., less than* $+\infty$*) is in* NP.

4.2 The regular average-value problem

In this section, we study the regular average-value problem for $\mathbf{VASS}(\mathbb{Z})$. We assume that $\mathcal{A}$ has a regular computation of finite value and does not have a regular computation of the value $-\infty$. Finally, we consider the case of the threshold $\lambda = 0$. The case of an arbitrary threshold $\lambda \in \mathbb{Q}$ can be easily reduced to the case $\lambda = 0$ (see [7, 6] for intuitions and techniques in mean-payoff games a.k.a. limit-average games).

Consider a regular computation π and let $\rho_1(\rho_2)^\omega$ be a path corresponding to π. We derive the necessary and sufficient conditions on ρ_1, ρ_2 to have $\textsc{LimAvg}(f(\pi)) \leq 0$.

Assume that $\textsc{LimAvg}(f(\pi)) \leq 0$. Due to Lemma 7, we have $\textsc{Gain}(\rho_2) \cdot \textsc{Vals}(\rho_2) = 0$, which implies that every iteration of ρ_2 has the same average. Therefore, $\textsc{LimAvg}(f(\pi))$ is the average value of ρ_2 (starting with counter values defined by ρ_1). The latter value is at most 0 if and only if the sum of values along ρ_2 is at most 0. We define this sum below.

The sum of values $\mathsf{Sum}_{\vec{g}}(\rho)$. Let l be the labeling defining the cost function f. Consider a path ρ of length m such that $\rho = (q_1, q_2, \vec{y}_1) \ldots (q_m, q_{m+1}, \vec{y}_m)$. Let π be the computation corresponding to ρ with the initial counter values being $\vec{g}$. Then, the value of counters at the position i in π is $\vec{g} + \sum_{j=1}^{i-1} \vec{y}_j$. Therefore, the sum of values over ρ starting with counter values $\vec{g} \in \mathbb{Z}^k$, denoted by $\mathsf{Sum}_{\vec{g}}(\rho)$, is given by the following formula:

$$\mathsf{Sum}_{\vec{g}}(\rho) = \sum_{i=1}^{m} \left(\vec{g} + \sum_{j=1}^{i-1} \vec{y}_j \right) \cdot l(q_i)$$

We have the following:

▶ **Lemma 10.** *There exists a regular computation π with $\textsc{LimAvg}(f(\pi)) \leq 0$ if and only if there exist paths ρ_1, ρ_2 such that*

(C1) *$\rho_1(\rho_2)^\omega$ is an infinite path and ρ_2 is a cycle,*

(C2) *$\textsc{Gain}(\rho_2) \cdot \textsc{Vals}(\rho_2) = 0$, and*

(C3) *$\mathsf{Sum}_{\textsc{Gain}(\rho_1)}(\rho_2) \leq 0$.*

Due to results from the previous section, we can check in NP the existence of ρ_1, ρ_2 satisfying (C1) and (C2). We call a cycle ρ_2 *balanced* if $\textsc{Gain}(\rho_2) \cdot \textsc{Vals}(\rho_2) = 0$. In the remaining part we focus on condition (C3), while keeping in mind (C1) and (C2).

The plan of the proof. The proof has three key ingredients which are as follows: *factorizations*, *quadratic factor elimination*, and *the linear case.*

- *Factorizations.* We show that we can consider only paths ρ_2 of the form

  $$\alpha_0 \beta_1^{n[1]} \alpha_1 \beta_2^{n[2]} \ldots \beta_p^{n[p]} \alpha_p,$$

  where $|\alpha_i|, |\beta_i| \leq |\mathcal{A}|$ and each β_i is a distinct simple cycle (Lemma 12). It follows that p is exponentially bounded in $|\mathcal{A}|$. Next, we show that for ρ_2 in such a form we have

  $$\mathsf{Sum}_{\textsc{Gain}(\rho_1)}(\rho_2) = \vec{n}^T \mathbf{B} \vec{n} + \vec{c} \cdot \vec{n} + e$$

  where $\mathbf{B} \in \mathbb{Z}^{p \times p}, \vec{c} \in \mathbb{Z}^p$ (Lemma 13). Therefore, $\mathsf{Sum}_{\textsc{Gain}(\rho_1)}(\rho_2) \leq 0$ can be presented as an instance of integer quadratic programming, where the variables correspond to multiplicities of simple cycles.

  However, there are two problems to overcome. First, ρ_2 has to be balanced, i.e., it has to satisfy $\textsc{Gain}(\rho_2) \cdot \textsc{Vals}(\rho_2) = 0$, which introduces another quadratic equation. Solving a system of two quadratic equations over integers is considerably more difficult (see [11] for references). Second, p is (bounded by) the number of distinct simple cycles and hence it can be exponential in $|\mathcal{A}|$. Therefore, $\mathbf{B}$ and $\vec{c}$ may have exponential size (of the binary representation) in $|\mathcal{A}|$.

- *Quadratic factor elimination.* To solve these problems, we fix a sequence $\textsc{Tpl} = (\alpha_0, \beta_1, \alpha_1, \ldots, \beta_p, \alpha_p)$ and consider $\mathbf{B}_{\textsc{Tpl}}, \vec{c}_{\textsc{Tpl}}, e_{\textsc{Tpl}}$ for $\textsc{Tpl}$. We show that one of the following holds (Lemma 14 and Lemma 15):

  - $\vec{n}^T \mathbf{B}_{\textsc{Tpl}} \vec{n} + \vec{c}_{\textsc{Tpl}} \cdot \vec{n} + e_{\textsc{Tpl}} \leq 0$ has a simple solution, or
  - there exist $\vec{d}_{\textsc{Tpl}} \in \mathbb{Z}^p, h_{\textsc{Tpl}} \in \mathbb{Z}$ such that for all vectors $\vec{n}$, if the cycle $\rho_2 = \alpha_0 \beta_1^{n[1]} \dots \beta_p^{n[p]} \alpha_p$ is balanced (C2), then $\vec{n}^T \mathbf{B}_{\textsc{Tpl}} \vec{n} = \vec{d}_{\textsc{Tpl}} \cdot \vec{n} + h_{\textsc{Tpl}}$.

  We can decide in non-deterministic polynomial time whether the first condition holds (Lemma 20).

- *The linear case.* Assuming that the second condition holds, we reduce the problem of solving the quadratic inequality $\vec{n}^T \mathbf{B}_{\textsc{Tpl}} \vec{n} + \vec{c}_{\textsc{Tpl}} \cdot \vec{n} + e_{\textsc{Tpl}} \leq 0$ to solving the linear inequality $(\vec{c}_{\textsc{Tpl}} + \vec{d}_{\textsc{Tpl}}) \cdot \vec{n} + (e_{\textsc{Tpl}} + h_{\textsc{Tpl}}) \leq 0$. Moreover, we can compute $\vec{d}_{\textsc{Tpl}}, h_{\textsc{Tpl}}$ from the sequence Tpl. At this point, we can solve the problem in non-deterministic exponential time. Next, we argue that we do not have to compute the whole system.

  We show that if $(\vec{c}_{\textsc{Tpl}} + \vec{d}_{\textsc{Tpl}}) \cdot \vec{n} + e_{\textsc{Tpl}} + h_{\textsc{Tpl}} \leq 0$ has a solution, then it has a solution for a vector $\vec{n}_0$ with $m = O(|\mathcal{A}|)$ non-zero components. Therefore, we can remove cycles corresponding to 0 coefficients of $\vec{n}_0$. Still, $\sum_{i=0}^{p} |\alpha_i|$ can be exponential in $|\mathcal{A}|$, but this operation shortens the size of the template, i.e., the value $\sum_{i=0}^{p} |\alpha_i| + \sum_{i=1}^{p} |\beta_i|$, and hence by iterating it we get a polynomial size template, which yields a polynomial-size system of inequalities. These inequalities can be solved in NP.

We now present the details of each ingredient.

4.2.1 Factorizations

The regular finite-value problem has been solved via reduction to solving (quadratic and linear) inequalities. In this section, we show a reduction of the regular average-value problem to linear and quadratic inequalities as well. First, we establish that we can consider only cycles ρ, which have a compact representation using *templates* parametrized by multiplicities of cycles. The value $\mathsf{Sum}_{\vec{g}}(\rho)$ for a cycle represented by a template is given by a quadratic function in the multiplicities of cycles.

Templates and multiplicities. A *template* Tpl is a sequence of paths $(\alpha_0, \beta_1, \alpha_1, \dots, \beta_p, \alpha_p)$ such that all $\beta_1, \dots, \beta_p$ are cycles and $\alpha_0 \beta_1 \alpha_1 \dots \beta_p \alpha_p$ is a cycle. A template is *minimal* if for all $i \in \{0, \dots, p\}$ we have $|\alpha_i| < |Q|$ and all β_i are pairwise distinct simple cycles. For every vector $\vec{n} \in \mathbb{N}^p$, called *multiplicities*, we define $\textsc{Tpl}(\vec{n})$ as a cycle $\alpha_0 \beta_1^{n[1]} \alpha_1 \beta_2^{n[2]} \dots \beta_p^{n[p]} \alpha_p$. A cycle ρ has a (minimal) *factorization* if there exists a (minimal) template and multiplicities $\vec{n}$ such that $\rho = \textsc{Tpl}(\vec{n})$.

Observe that every cycle ρ has a factorization such that for all i we have $|\alpha_i| < |Q|$ and each β_i is a simple cycle. However, the sequence β_i's can have repetitions. The following lemma states that if a cycle β occurs twice in ρ, then we can group them together.

▶ **Lemma 11.** *Consider $\vec{g} \in \mathbb{Z}^k$ and a cycle $\alpha_0 \beta \alpha_1 \beta \alpha_2$. Then, one of the following holds:* $\mathsf{Sum}_{\vec{g}}(\alpha_0 \beta^2 \alpha_1 \alpha_2) \leq \mathsf{Sum}_{\vec{g}}(\alpha_0 \beta \alpha_1 \beta \alpha_2)$ *or* $\mathsf{Sum}_{\vec{g}}(\alpha_0 \alpha_1 \beta^2 \alpha_2) \leq \mathsf{Sum}_{\vec{g}}(\alpha_0 \beta \alpha_1 \beta \alpha_2)$.

Careful repeated application of Lemma 11 implies that we can look for a cycle ρ satisfying (C1), (C2) and (C3) among cycles that have a minimal factorization:

▶ **Lemma 12.** *For every cycle ρ and $\vec{g} \in \mathbb{Z}^k$, there exists a cycle ρ' that has a minimal factorization such that* $\textsc{Gain}(\rho) = \textsc{Gain}(\rho')$, $\textsc{Vals}(\rho) = \textsc{Vals}(\rho')$ *and* $\mathsf{Sum}_{\vec{g}}(\rho) \geq \mathsf{Sum}_{\vec{g}}(\rho')$.

Consider a template $\textsc{Tpl} = (\alpha_0, \beta_1, \alpha_1, \dots, \beta_p, \alpha_p)$. We present $\mathsf{Sum}_{\vec{g}}(\textsc{Tpl}(\vec{n}))$ as a function from multiplicities of simple cycles $\vec{n} \in \mathbb{N}$ into $\mathbb{Z}$. First, observe that

$$\mathsf{Sum}_{\vec{g}}(\textsc{Tpl}(\vec{n})) = \mathsf{Sum}_{\vec{0}}(\textsc{Tpl}(\vec{n})) + \vec{g} \cdot \textsc{Vals}(\textsc{Tpl}(\vec{n})).$$

The expression $\textsc{Vals}(\textsc{Tpl}(\vec{n}))$ is a linear expression in $\vec{n}$ with natural coefficients, and the expression $\mathsf{Sum}_{\vec{0}}(\textsc{Tpl}(\vec{n}))$ is a quadratic function in each of its arguments $\vec{n}$.

▶ **Lemma 13.** *Given a template* $\textsc{Tpl}$ *we can compute in polynomial time in* $|\textsc{Tpl}|+|\mathcal{A}|+|f|$, *a symmetric matrix* $\mathbf{B}_{\textsc{Tpl}} \in \mathbb{Z}^{p\times p}, \vec{c}_{\textsc{Tpl}} \in \mathbb{Z}^p$ *and* $e_{\textsc{Tpl}} \in \mathbb{Z}$ *such that the following holds:*

$$2 \cdot \mathsf{Sum}_{\vec{0}}(\textsc{Tpl}(\vec{n})) = \vec{n}^T \mathbf{B}_{\textsc{Tpl}}\vec{n} + \vec{c}_{\textsc{Tpl}}\vec{n} + e_{\textsc{Tpl}} \tag{3}$$

Moreover, for all $i, j \in \{1, \ldots, p\}$ *we have*

$$\mathbf{B}_{\textsc{Tpl}}[i,j] = \textsc{Gain}(\beta_{min(i,j)}) \cdot \textsc{Vals}(\beta_{max(i,j)}). \tag{4}$$

Observe that $\mathbf{B}_{\textsc{Tpl}}$ is similar to the matrix $\mathbf{A}$ from (1). We exploit this similarity in the following section to eliminate the term $\vec{n}^T\mathbf{B}_{\textsc{Tpl}}\vec{n}$ if possible.

4.2.2 Elimination of the quadratic factor

We show how to simplify the expression (3) of Lemma 13 for $\mathsf{Sum}_{\vec{0}}(\textsc{Tpl}(\vec{n}))$. We show that either the inequality $\mathsf{Sum}_{\vec{0}}(\textsc{Tpl}(\vec{n})) + \vec{g} \cdot \textsc{Vals}(\textsc{Tpl}(\vec{n})) \leq 0$ has a simple solution for every $\vec{g}$, or the quadratic term in (3) of Lemma 13 can be substituted with a linear term.

Negative and linear templates. Consider a template $\textsc{Tpl}$. A template $\textsc{Tpl}$ is *positive* (resp., *negative*) if there exist multiplicities $\vec{n}_1, \vec{n}_2 \in \mathbb{N}^p$ such that

1. $\vec{n}_1^T\mathbf{B}_{\textsc{Tpl}}\vec{n}_1 > 0$ (resp., $\vec{n}_1^T\mathbf{B}_{\textsc{Tpl}}\vec{n}_1 < 0$), and
2. for every $t \in \mathbb{N}^+$, we have $\textsc{Gain}(\textsc{Tpl}(t\vec{n}_1 + \vec{n}_2)) \cdot \textsc{Vals}(\textsc{Tpl}(t\vec{n}_1 + \vec{n}_2)) = 0$.

A template $\textsc{Tpl}$ is *linear* if there exist $\vec{d}_{\textsc{Tpl}} \in \mathbb{Z}^p$ and $h_{\textsc{Tpl}} \in \mathbb{Z}$ such that for all $\vec{n}$, if $\textsc{Gain}(\textsc{Tpl}(\vec{n})) \cdot \textsc{Vals}(\textsc{Tpl}(\vec{n})) = 0$, then $\vec{n}^T\mathbf{B}_{\textsc{Tpl}}\vec{n} = \vec{d}_{\textsc{Tpl}} \cdot \vec{n} + h_{\textsc{Tpl}}$.

We observe that the existence of a negative cycle $\textsc{Tpl}$ implies that $\mathsf{Sum}_{\vec{g}}(\textsc{Tpl}(\vec{n})) \leq 0$ has a solution for every $\vec{g} \in \mathbb{N}^k$, which in turn implies that the answer to the average-value problem is YES. Basically, for ρ^t defined as $\textsc{Tpl}(t\vec{n}_1 + \vec{n}_2)$ and t big enough we can make $\mathsf{Sum}_{\vec{g}}(\rho^t)$ arbitrarily small.

▶ **Lemma 14.** *If there exist a negative template, whose any state is reachable from some initial state, then the answer to the regular average-value problem with threshold* 0 *is YES.*

We show that either there is a template, which is negative or all templates are linear (Lemma 15). Next, we show that we can check in non-deterministic polynomial time whether there exists a negative template (Lemma 20).

▶ **Lemma 15.** *(1) There exists a negative template or all templates are linear. (2) If there exists a positive template* $\textsc{Tpl}$, *then there exists a negative one of the size bounded by* $|\textsc{Tpl}|^2$. *(3) If a template* $\textsc{Tpl}$ *is linear, we can compute* $\vec{d}_{\textsc{Tpl}}, h_{\textsc{Tpl}}$ *in polynomial time in* $|\textsc{Tpl}| + |\mathcal{A}| + |f|$.

Proof ideas. Consider a template $\textsc{Tpl}$ with all connecting paths being empty, i.e., $\textsc{Tpl} = (\epsilon, \beta_1, \ldots, \beta_p, \epsilon)$. Let $\vec{n}$ be a vector of multiplicities such that a cycle $\textsc{Tpl}(\vec{n})$ is balanced, i.e., $\textsc{Gain}(\textsc{Tpl}(\vec{n})) \cdot \textsc{Vals}(\textsc{Tpl}(\vec{n})) = 0$. We consider three cases:

The case $\vec{n}^T\mathbf{B}_{\mathsf{Tpl}}\vec{n} < 0$. Then, $\vec{n}_1 = \vec{n}$ and $\vec{n}_2 = \vec{0}$ witness negativity of $\textsc{Tpl}$.

The case $\vec{n}^T\mathbf{B}_{\mathsf{Tpl}}\vec{n} > 0$. Then, $\textsc{Tpl}$ is positive and we show that the reversed template $\textsc{Tpl}^R = (\epsilon, \beta_p, \ldots, \beta_1, \epsilon)$ is negative. Equality $\textsc{Gain}(\textsc{Tpl}(\vec{n})) \cdot \textsc{Vals}(\textsc{Tpl}(\vec{n})) = 0$ can be stated as a matrix equation $\vec{n}^T\mathbf{A}\vec{n} = 0$, where $\mathbf{A}$ is defined as in (2). We juxtapose (2) and (4), and get that for all i, j we have $\mathbf{A}[i,j] = \mathbf{B}_{\textsc{Tpl}}[i,j] + \mathbf{B}_{\textsc{Tpl}^R}[p-i+1, p-j+1]$. Thus, $0 = \vec{n}^T\mathbf{A}\vec{n} = \vec{n}^T\mathbf{B}_{\textsc{Tpl}}\vec{n} + \vec{n_R}^T\mathbf{B}_{\textsc{Tpl}^R}\vec{n_R}$, where $\vec{n}_R$ is the reversed vector $\vec{n}$. It follows that $\textsc{Tpl}^R$ is negative.

The above two cases fail. Then for all $\vec{n}$, if $\textsc{Gain}(\textsc{Tpl}(\vec{n})) \cdot \textsc{Vals}(\textsc{Tpl}(\vec{n})) = 0$, then $\vec{n}^T \mathbf{B}_{\textsc{Tpl}} \vec{n} = 0$. Therefore, $\textsc{Tpl}$ is linear with $\vec{d}_{\textsc{Tpl}}, h_{\textsc{Tpl}}$ being $\vec{0}$ and 0 respectively.

Essentially the same line of reasoning can be applied if the connecting paths in $\textsc{Tpl}$ are non-empty. However, in that case $\vec{d}_{\textsc{Tpl}}, h_{\textsc{Tpl}}$ may be non-zero. ◀

Now, we discuss how to check whether there exists a negative template. One of the conditions of negativity is that $\textsc{Gain}(\textsc{Tpl}(t\vec{n}_1 + \vec{n}_2)) \cdot \textsc{Vals}(\textsc{Tpl}(t\vec{n}_1 + \vec{n}_2)) = 0$. Recall that $\textsc{Gain}$ and $\textsc{Vals}$ depend on the multiset of transitions, but not on the order of transitions. Therefore, for a given template $\textsc{Tpl}$, we define $\textsc{Trans}_{\textsc{Tpl}}(\vec{n}) \in \mathbb{N}^m$, where $m = |\delta|$, as the vector of multiplicities of transitions in $\textsc{Tpl}(\vec{n})$. We will write $\textsc{Trans}(\vec{n})$ if $\textsc{Tpl}$ is clear from the context.

▶ **Lemma 16.** *Let* $\textsc{Tpl}$ *be a template and let* $\vec{n} \in \mathbb{N}^p$ *be a vector of multiplicities. There exist* $r_1, \ldots, r_\ell \in \mathbb{Q}^+$ *and* $\vec{z}_1, \ldots, \vec{z}_\ell \in \mathbb{N}^p$ *such that (1) supp*$(\vec{z}_i) \leq m$ *(the number of transitions of* $|\mathcal{A}|$*), (2) there exists* $t \in \mathbb{N}^+$ *such that* $\textsc{Trans}(\vec{z}_i) = t \cdot \textsc{Trans}(\vec{n})$*, and (3)* $\vec{n} = \sum_{i=1}^{\ell} r_i \vec{z}_i$.

▶ Remark. The condition $\textsc{Trans}(\vec{z}_i) = t \cdot \textsc{Trans}(\vec{n})$ implies that if the cycle $\textsc{Tpl}(\vec{n})$ is balanced, then all the cycles $\textsc{Tpl}(\vec{z}_1), \ldots, \textsc{Tpl}(\vec{z}_\ell)$ are balanced as well.

Proof ideas. First, observe that $\textsc{Trans}$ is a linear function transforming vectors from $\mathbb{N}^p$ into vectors from $\mathbb{N}^m$. The value p can be exponential w.r.t. $m = |\delta|$ and hence we show that each vector from $\mathbb{N}^p$ can be presented as a linear combination over $\mathbb{Q}^+$ of vectors with polynomially-bounded supports. ◀

Next, we show that if there exists a negative template, then there exists one of polynomial size. If $\vec{n}_1, \vec{n}_2 \in \mathbb{N}^p$ are the vectors witnessing positivity (resp., negativity) of a template $\textsc{Tpl}$, then by Lemma 16, there exist witnesses $\vec{n}_1^0, \vec{n}_2^0$ with polynomial-size support. We remove from $\textsc{Tpl}$ cycles corresponding to coefficient 0 in both $\vec{n}_1^0$ and $\vec{n}_2^0$ and obtain only polynomially many cycles. In consequence, we have:

▶ **Lemma 17.** *If there exists a negative template, then there exists one of polynomial size in* $|\textsc{Tpl}| + |\mathcal{A}| + |f|$.

Still, to check whether there exists a negative template we have to solve a system consisting of a quadratic inequality $\vec{n}_1^T \mathbf{B}_{\textsc{Tpl}} \vec{n}_1 < 0$ and a quadratic equation, which corresponds to $\textsc{Gain}(\textsc{Tpl}(t\vec{n}_1 + \vec{n}_2)) \cdot \textsc{Vals}(\textsc{Tpl}(t\vec{n}_1 + \vec{n}_2)) = 0$. We show that this quadratic equation can be transformed into a system of linear inequalities. We show that using standard elimination of quadratic terms for successive variables $n[1], n[2], \ldots, n[p+1]$. The key observation is that $\textsc{Gain}(\textsc{Tpl}(\vec{n})) \cdot \textsc{Vals}(\textsc{Tpl}(\vec{n})) = 0$ implies that for every variable $n[i]$ either $n[i] = 0$ or $n[i]$ is a double root ($\Delta = 0$). Thanks to this property, we obtain a polynomial-size system of linear inequalities.

▶ **Lemma 18.** *Let* $\textsc{Tpl}$ *be a template. There exist systems of linear equations and inequalities* $S_1, \ldots, S_l$ *such that (1) each* S_i *has polynomial size in* $|\textsc{Tpl}| + |\mathcal{A}| + |f|$*, (2) for all* $\vec{n} \in \mathbb{N}^p$ *we have* $\textsc{Gain}(\textsc{Tpl}(\vec{n})) \cdot \textsc{Vals}(\textsc{Tpl}(\vec{n})) = 0$ *iff for some* i *the vector* $\vec{n}$ *satisfies* S_i.

Finally, Lemma 17 and Lemma 18 imply that existence of a negative template can be solved in NP via reduction to integer quadratic programming.

▶ **Lemma 19.** *We can verify in* NP *whether a given* VASS$(\mathbb{Z}, k)$ $\mathcal{A}$ *has a negative template.*

Proof sketch. We non-deterministically pick a template $\textsc{Tpl}$ of polynomial size (Lemma 17). Then, we non-deterministically pick a system S_i for template (Lemma 18). We make two copies of S_i: S_i^1 and S_i^2; in one we substitute $\vec{n}$ with $\vec{n}_2$ and in the other with $\vec{n}_1 + \vec{n}_2$. It follows that for all $t \in \mathbb{R}$ the vector $t\vec{n}_1 + \vec{n}_2$ satisfies S_i (substituted for $\vec{n}$). Finally, we solve an instance of integer quadratic programming consisting of $\vec{n}_1 \mathbf{B}_{\textsc{Tpl}} \vec{n}_1 - 1 \leq 0$ and linear equations and inequalities S_i^1 and S_i^2, which is an instance of integer quadratic programming and hence can be solved in NP [11]. ◀

4.2.3 The linear case

We consider the final case, where all templates are linear. The decision procedure described in the following lemma answers YES (in at least one of non-deterministic computations) whenever the answer to the regular average-value problem with threshold 0 is YES and all templates are linear. Thus, it is complete.

Our main algorithm assumes that all templates are linear if it fails to find a negative template. The failure can be due to a wrong non-deterministic pick. Having that in mind, we make sure that the decision procedure from the following lemma is sound regardless of the linearity of templates, i.e., if it answers YES, then the answer to the regular average-value problem with threshold 0 is YES.

▶ **Lemma 20.** *Assume that all templates are linear. Then, we can solve the regular average-value problem with threshold* 0 *in non-deterministic polynomial time. Moreover, the procedure is sound irrespectively of the linearity assumption.*

Proof sketch. First, we show that using Lemma 16, if there is a cycle ρ with (a) $\textsc{Gain}(\rho) \cdot \textsc{Vals}(\rho) = 0$ and (b) $\mathsf{Sum}_{\vec{y}}(\rho) \leq 0$, then there is a cycle ρ' defined by a template of polynomial size satisfying both conditions (a) and (b). We take a path ρ, write it as $\textsc{Tpl}(\vec{n})$ and apply Lemma 16 to $\vec{n}$. We get a vector with polynomially many non-zero coefficients $\vec{n}_0$ and define a reduced template $\textsc{Tpl}'$ by removal of cycles that correspond to 0 coefficients.

Second, consider a template $\textsc{Tpl}$ of polynomial size. We nondeterministically pick a subset of states Q and write a system of equations $S^Q_{\textsc{Gain}}$ over variables $\vec{x}, \vec{y}$ such that $\vec{x}, \vec{y}$ is a solution of $S^Q_{\textsc{Gain}}$ if and only if there exists a path ρ_1 satisfying (a) $\vec{x}$ are multiplicities of transitions along ρ_1 (b) ρ_1 is from some initial state of $\mathcal{A}$ to the first state of $\textsc{Tpl}$, (c) ρ_1 visits all states from Q, (d) $\vec{y} = \textsc{Gain}(\rho_1)$.

Finally, if all templates are linear, then there exist ρ_1, ρ_2 defining a regular computation of the average value at most 0 if and only if there is a subset of states Q and a template $\textsc{Tpl}$ of polynomial size such that the system of inequalities consisting of $S^Q_{\textsc{Gain}}$ and the inequality $H_{x,y} : (\vec{c}_{\textsc{Tpl},\vec{0}} + \vec{d}_{\textsc{Tpl}}) \cdot \vec{n} + e_{\textsc{Tpl},\vec{0}} + h_{\textsc{Tpl}} + 2 \cdot \vec{y} \cdot \textsc{Vals}(\textsc{Tpl}(\vec{n})) \leq 0$ has a solution over natural numbers. Note that all the components except for $\vec{y} \cdot \textsc{Vals}(\textsc{Tpl}(\vec{n}))$ are linear. Since $\vec{y}$ and $\vec{n}$ are variables, the component $\vec{y} \cdot \textsc{Vals}(\textsc{Tpl}(\vec{n}))$ is quadratic. Still, $S^Q_{\textsc{Gain}}$ with $H_{x,y}$ is an instance of integer quadratic programming, which can be solved in NP [11].

Having a solution of the system $S^Q_{\textsc{Gain}}$, we can compute in polynomial time $\textsc{Gain}(\rho_1)$ and $\mathsf{Sum}_{\textsc{Gain}(\rho_1)}(\textsc{Tpl}(\vec{n}))$ and then verify $\mathsf{Sum}_{\textsc{Gain}(\rho_1)}(\textsc{Tpl}(\vec{n})) \leq 0$. Therefore, the correctness of the algorithm does not depend on the linearity assumption. ◀

4.2.4 Summary

We present a short summary of the non-deterministic procedure deciding whether a given VASS$(\mathbb{Z}, k)$ $\mathcal{A}$ has a regular computation of the value at most 0. We assume that all states in $\mathcal{A}$ are reachable from initial states.

- *Step 1.* Check whether there is a cycle ρ with $\textsc{Gain}(\rho) \cdot \textsc{Vals}(\rho) < 0$. It can be done in non-deterministic polynomial time (Lemma 9). If the answer is YES, then the answer to the average-value problem is YES (Lemma 7). Otherwise, proceed to Step 2.
- *Step 2.* Check whether there exists a negative template in $\mathcal{A}$. It can be done in non-deterministic polynomial time (Lemma 19). If the answer is YES, then the answer to the regular average-value problem is YES (Lemma 14). Otherwise, proceed to Step 3.
- *Step 3.* Assuming that the previous steps failed, all templates are linear. Solve the regular average-value problem in non-deterministic polynomial time (Lemma 20). Note that Step 2, could have failed due to unfortunate non-deterministic pick. However, the procedure from Lemma 20 is sound regardless of the linearity assumption.

In consequence, we have the main result of this section:

▶ **Lemma 21.** *The regular average-value problem for* ***VASS***$(\mathbb{Z})$ *with (general) cost functions is in* NP.

4.3 Hardness

We show that the regular finite-value and the regular average-value problems are NP-hard. The proof is via reduction from the 3-SAT problem. Given a 3-CNF formula φ over n variables, we construct a VASS of dimension $2n$, where dimensions correspond to literals in φ. Each simple cycle ρ in the VASS consists of two parts: The first part corresponds to picking a substitution σ, which is stored in the vector $\textsc{Gain}(\rho)$. The second part ensures that $\textsc{Gain}(\rho) \cdot \textsc{Vals}(\rho) = 0$ if σ satisfies φ and it is strictly positive otherwise. Therefore, if φ is satisfiable, the VASS has a regular run of the average cost 0, and otherwise all its regular runs have infinite average cost. In consequence, we have the following:

▶ **Lemma 22.** *The regular finite-value and the regular average-value problems for* ***VASS***$(\mathbb{Z})$ *with (general) cost functions are* NP*-hard.*

The main results of this section (Lemma 9, Lemma 21 and Lemma 22) summarize to Theorem 6. We leave the case of non-regular runs as an open question.

▶ **Open question 23.** *What is the complexity of the average-value and finite-value problems for* ***VASS***$(\mathbb{Z})$*?*

5 General cost functions and VASS$(\mathbb{N})$

We show that the average-value and the regular average-value problems are undecidable. The proofs are via reduction from the halting problem for Minsky machines [17], which are automata with two natural-valued *registers* r_1, r_2. There are two main differences between Minsky machines and **VASS**$(\mathbb{N})$. First, the former can perform zero- and nonzero-tests on their registers, while the latter can take any transition as long as the counters' values remain non-negative. Second, the halting problem for Minsky machines is qualitative, i.e., the answer is YES or NO. We consider quantitative problems for VASS, where we are interested in the values assigned to computations. We exploit the quantitative features of our problems to simulate zero- and nonzero-tests.

The problems with an exact threshold are undecidable, but we can decide existence of a regular computation of some finite value via reduction to reachability in VASS.

▶ **Theorem 24.** *(1) The average-value and the regular average-value problems for* ***VASS***$(\mathbb{N})$ *with (general) cost functions are undecidable. (2) The regular finite-value problem for* ***VASS***$(\mathbb{N})$ *with (general) cost functions is decidable.*

References

1 Parosh Aziz Abdulla, Mohamed Faouzi Atig, Piotr Hofman, Richard Mayr, K. Narayan Kumar, and Patrick Totzke. Infinite-state energy games. In *CSL-LICS 2014*, pages 7:1–7:10, 2014.

2 Roderick Bloem, Swen Jacobs, Ayrat Khalimov, Igor Konnov, Sasha Rubin, Helmut Veith, and Josef Widder. Decidability in Parameterized Verification. *SIGACT News*, 47(2):53–64, 2016.

3 Patricia Bouyer, Nicolas Markey, Mickael Randour, Kim G. Larsen, and Simon Laursen. Average-energy games. *Acta Inf.*, 55(2):91–127, 2018.

4 Tomás Brázdil, Krishnendu Chatterjee, Antonín Kucera, Petr Novotný, Dominik Velan, and Florian Zuleger. Efficient Algorithms for Asymptotic Bounds on Termination Time in VASS. In *LICS 2018*, pages 185–194, 2018.

5 Tomás Brázdil, Stefan Kiefer, Antonín Kucera, and Petr Novotný. Long-Run Average Behaviour of Probabilistic Vector Addition Systems. In *LICS 2015*, pages 44–55, 2015. doi:10.1109/LICS.2015.15.

6 Krishnendu Chatterjee, Thomas A. Henzinger, and Jan Otop. Nested Weighted Limit-Average Automata of Bounded Width. In *MFCS 2016*, pages 24:1–24:14, 2016.

7 Krishnendu Chatterjee, Thomas A. Henzinger, and Jan Otop. Quantitative Monitor Automata. In *SAS 2016*, pages 23–38, 2016.

8 Krishnendu Chatterjee and Yaron Velner. Hyperplane separation technique for multidimensional mean-payoff games. *J. Comput. Syst. Sci.*, 88:236–259, 2017.

9 Krishnendu Chatterjee and Yaron Velner. The Complexity of Mean-Payoff Pushdown Games. *J. ACM*, 64(5):34:1–34:49, 2017.

10 Wojciech Czerwinski, Slawomir Lasota, Ranko Lazic, Jérôme Leroux, and Filip Mazowiecki. The Reachability Problem for Petri Nets is Not Elementary. In *STOC*, pages 398–406, 2019.

11 Alberto Del Pia, Santanu S Dey, and Marco Molinaro. Mixed-integer quadratic programming is in NP. *Mathematical Programming*, 162(1-2):225–240, 2017.

12 Emanuele D'Osualdo, Jonathan Kochems, and C. H. Luke Ong. Automatic Verification of Erlang-Style Concurrency. In Francesco Logozzo and Manuel Fähndrich, editors, *SAS 2013*, pages 454–476, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. doi:10.1007/978-3-642-38856-9_24.

13 Javier Esparza. Decidability and complexity of Petri net problems—an introduction. *Lectures on Petri nets I: Basic models*, pages 374–428, 1998.

14 Javier Esparza and Mogens Nielsen. Decidability Issues for Petri Nets - a survey. *Bulletin of the European Association for Theoretical Computer Science*, 52:245–262, 1994.

15 Yu Feng, Ruben Martins, Yuepeng Wang, Isil Dillig, and Thomas W. Reps. Component-based Synthesis for Complex APIs. In *POPL 2017*, POPL 2017, pages 599–612, New York, NY, USA, 2017. ACM. doi:10.1145/3009837.3009851.

16 Pierre Ganty and Rupak Majumdar. Algorithmic Verification of Asynchronous Programs. *ACM Trans. Program. Lang. Syst.*, 34(1):6:1–6:48, May 2012. doi:10.1145/2160910.2160915.

17 John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.

18 Alexander Kaiser, Daniel Kroening, and Thomas Wahl. Dynamic Cutoff Detection in Parameterized Concurrent Programs. In *CAV 2010*, pages 645–659, 2010. doi:10.1007/978-3-642-14295-6_55.

19 Alexander Kaiser, Daniel Kroening, and Thomas Wahl. Efficient Coverability Analysis by Proof Minimization. In Maciej Koutny and Irek Ulidowski, editors, *CONCUR 2012*, pages 500–515, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. doi:10.1007/978-3-642-32940-1_35.

20 Richard M. Karp and Raymond E. Miller. Parallel Program Schemata. *J. Comput. Syst. Sci.*, 3(2):147–195, 1969.

21 S. Rao Kosaraju. Decidability of Reachability in Vector Addition Systems (Preliminary Version). In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, pages 267–281, 1982. doi:10.1145/800070.802201.

22 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`.

23 Jérôme Leroux. Vector addition systems reachability problem (a simpler solution). In *EPiC*, volume 10, pages 214–228. Andrei Voronkov, 2012.

24 Jérôme Leroux. Polynomial Vector Addition Systems With States. In *ICALP 2018*, pages 134:1–134:13, 2018.

25 R. Lipton. The Reachability Problem Requires Exponential Space. Technical report 62, Yale, 1976.

26 Ernst W. Mayr. An Algorithm for the General Petri Net Reachability Problem. In *STOC 1981*, pages 238–246, 1981. `doi:10.1145/800076.802477`.

27 Jakub Michaliszyn and Jan Otop. Average Stack Cost of Büchi Pushdown Automata. In *FSTTCS 2017*, pages 42:1–42:13, 2017. `doi:10.4230/LIPIcs.FSTTCS.2017.42`.

28 Charles Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6(2):223–231, 1978. `doi:10.1016/0304-3975(78)90036-1`.

29 Moritz Sinn, Florian Zuleger, and Helmut Veith. A Simple and Scalable Static Analysis for Bound Analysis and Amortized Complexity Analysis. In *CAV*, pages 745–761, 2014.