Deciding Termination of Simple Randomized Loops

Éléanore Meyer
RWTH Aachen University, Germany

Jürgen Giesl
RWTH Aachen University, Germany

Abstract

We show that universal positive almost sure termination (UPAST) is decidable for a class of simple randomized programs, i.e., it is decidable whether the expected runtime of such a program is finite for all inputs. Our class contains all programs that consist of a single loop, with a linear loop guard and a loop body composed of two linear commuting and diagonalizable updates. In each iteration of the loop, the update to be carried out is picked at random, according to a fixed probability. We show the decidability of UPAST for this class of programs, where the program's variables and inputs may range over various sub-semirings of the real numbers. In this way, we extend a line of research initiated by Tiwari in 2004 into the realm of randomized programs.

2012 ACM Subject Classification Theory of computation → Probabilistic computation; Theory of computation → Program analysis

Keywords and phrases decision procedures, randomized programs, linear loops, positive almost sure termination

Digital Object Identifier 10.4230/LIPIcs.MFCS.2025.76

Related Version *Full version, including all proofs*: `https://arxiv.org/abs/2506.18541` [26]

Supplementary Material
Software (Source Code): `https://github.com/aprove-developers/SiRop` [24]
archived at `swh:1:dir:3ca664cebef79bfeb95ec944ddc8441d3b528bf6`

Funding DFG Research Training Group 2236 UnRAVeL.

Acknowledgements We thank Sophia Greiwe for her help with the implementation of our decision procedure in SiRop.

1 Introduction

We consider the problem of universal positive almost sure termination (UPAST), i.e., deciding whether a given randomized program has finite expected runtime on all inputs [6, 34]. This is a stronger property than universal almost sure termination (UAST) which requires that the probability of termination is 1. Our programs are simple randomized loops of the form

$$\texttt{while } \mathbf{C}\vec{x} > \vec{0} : \vec{x} \leftarrow \mathbf{A}\vec{x} \oplus_p \mathbf{B}\vec{x} \qquad (1)$$

Here, $\vec{x} = (x_1, \ldots, x_n)$ denotes the vector of program variables that range over a semiring $\mathcal{S} \subseteq \mathbb{R}$, and $\mathbf{C} \in \mathbb{R}^{m\times n}$ is a matrix representing the loop guard with m linear constraints over the program variables. In each execution of the loop body, a matrix is chosen among $\mathbf{A}, \mathbf{B} \in \mathcal{S}^{n\times n}$ according to the probability $p \in [0, 1]$ and the value $\vec{x}$ is updated accordingly.

50th International Symposium on Mathematical Foundations of Computer Science (MFCS 2025).
Editors: Paweł Gawrychowski, Filip Mazowiecki, and Michał Skrzypczak; Article No. 76; pp. 76:1–76:19
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Our Contribution. We show that UPAST is decidable for all $\mathcal{S} \in \{\mathbb{Z}, \mathbb{Q}, \mathbb{A}\}$ when limited to loops with diagonalizable commuting matrices $\mathbf{A}$ and $\mathbf{B}$, where $\mathbb{A}$ is the set of algebraic real numbers.[1] Thus, we extend previous results on the termination of linear and affine[2] non-randomized loops to the randomized setting. In addition to deciding universal termination, our approach can compute a non-termination witness $\vec{x} \in \mathcal{S}^n$, i.e., if the loop is non-terminating, then $\vec{x}$ is an input leading to an infinite expected runtime.

Our programs go beyond single path loops as we might have $\mathbf{A} \neq \mathbf{B}$. Thus, for every $k \in \mathbb{N}$, there is not just a single execution of length k but one has a "range" of possible executions of length k where each execution occurs according to a known probability. To ensure tractability of the resulting problem we require commutativity of both updates, so that we can focus on how often each update has been selected in an execution, but we do not have to take the 2^k different orders into account in which the two updates might have been chosen. Moreover, we require diagonalizability to obtain closed forms of a certain shape, which allows us to analyze the behavior of a "range" of different executions at once. To demonstrate the practical applicability of our decision procedure and the computation of non-termination witnesses, we provide a prototype implementation for the case $\mathcal{S} = \mathbb{A}$ with our tool SiRop.

Related Work. We continue a line of research started in 2004 by Tiwari [35] who showed decidability of universal termination for loops with an affine guard and an affine update as its body, where the guard, updates, and inputs range over the real numbers. In his proof, Tiwari reduced the affine to the linear case. In 2006, Braverman [7] proved that the problem remains decidable for loops and inputs ranging over the rational numbers $\mathbb{Q}$, and if the guard and update are linear, then he also showed decidability over the integers $\mathbb{Z}$. Similar to Tiwari, Braverman also reduced the affine case for $\mathbb{Q}$ to the linear case. In 2015, Ouaknine et al. proved [31] that the affine case is decidable over the integers $\mathbb{Z}$ whenever the update is of the form $\vec{x} \leftarrow \mathbf{A}\vec{x} + \vec{a}$, provided that $\mathbf{A} \in \mathbb{Z}^{n\times n}$ is diagonalizable. This restriction was removed by Hosseini et al. in 2019 [18]. In a related line of work, we proved decidability of universal termination over the integers $\mathbb{Z}$ for triangular affine loops, i.e., where the matrix $\mathbf{A} \in \mathbb{Z}^{n\times n}$ is triangular [12]. Later, we extended these results to triangular weakly non-linear loops which extend triangular loops by allowing certain non-linear updates [15, 16].

The only decidability results for termination of randomized programs that we are aware of consider probabilistic vector addition systems [8] or constant probability programs [13], i.e., loops whose guards consist of only one affine inequation and whose bodies consist of several probabilistic branches (with fixed probabilities) that may increase or decrease the program variables by fixed constants. The programs in [8, 13] are orthogonal to the ones considered in our approach as they only allow to modify the program variables by adding constants, but do not allow for multiplication. Another related area of research [4, 21, 28] deals with prob-solvable loops and moment invariants. Given such a loop, these techniques can compute closed forms for all moments of program variables for a given iteration of the loop and, by taking a limit, also upon the loop's termination. Thus, if restricted to almost surely terminating programs, they can decide UPAST. However, in contrast to our method, these techniques require that all variables in the loop guard may only take finitely many values. Moreover, there are many automated approaches for tackling UPAST using so-called ranking supermartingales (RSM), e.g., [1, 3, 5, 9, 10, 25, 27, 30, 36]. To generate a suitable RSM, one

---

[1] Our approach only considers *algebraic* real p, $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, as it is not possible to represent arbitrary real numbers on computers. However, in Sect. 6 we will see that such a loop terminates for all algebraic real inputs iff it terminates for all real inputs.

[2] In an affine (or non-homogeneous) loop, the guard may have the form $\mathbf{C}\vec{x} > \vec{c}$ and the update may have the form $\vec{x} \leftarrow \mathbf{A}\vec{x} + \vec{a}$ for arbitrary vectors $\vec{c}$ and $\vec{a}$.

often uses techniques based on affine or polynomial templates, which renders the approach incomplete. In [19], the authors showed that deciding UPAST is harder than deciding universal termination for non-randomized programs in terms of the arithmetic hierarchy.

Outline of our Approach. We assume familiarity with basics from probability theory and linear algebra (we recapitulate some main concepts in [26]). Sect. 2 formally introduces simple randomized loops, gives their semantics in terms of a probability space, and presents most of the definitions used throughout the paper. For a loop as in (1), we consider (finite) executions f corresponding to words over the alphabet $\{A, B\}$, where the i-th symbol in f indicates which update matrix was used in the i-th application of the assignment $\vec{x} \leftarrow \mathbf{A}\vec{x} \oplus_p \mathbf{B}\vec{x}$. For such executions f, $|f|_A$ and $|f|_B$ denote the number of A- and B-symbols in f, respectively. Moreover, we introduce the function $\mathcal{V}al_{\vec{x}}$ that maps finite executions f to the values $\mathcal{V}al_{\vec{x}}(f) \in \mathbb{R}^m$ of the constraints in the loop guard after executing f on a concrete input $\vec{x} \in \mathbb{R}^n$, i.e., $\mathcal{V}al_{\vec{x}}(f) = \mathbf{C} \cdot \mathbf{A}^{|f|_A} \cdot \mathbf{B}^{|f|_B} \cdot \vec{x}$, since $\mathbf{A}$ and $\mathbf{B}$ commute. Our decision procedure does not search for a non-terminating input directly, but for an *eventually* non-terminating input $\vec{x}$. An input $\vec{x}$ is eventually non-terminating if by repeated execution of the loop body on $\vec{x}$ (while ignoring the guard), a non-terminating input can eventually be reached. In Lemma 12, we show that a loop has an eventually non-terminating input iff it also has a non-terminating input. Later (in Sect. 6) we will show how to lift such an eventually non-terminating input to an actual non-terminating input.

In Sect. 3, we introduce a mapping $\mathcal{U}$ that maps executions f to the difference between the relative number $\frac{|f|_A}{|f|}$ of times that the update matrix $\mathbf{A}$ has been chosen in f and the probability p of choosing $\mathbf{A}$, i.e., $\mathcal{U}(f) = \frac{|f|_A}{|f|} - p$. Moreover, we essentially partition the set of indices $\{1, \ldots, n\}$ of all program variables into suitable sets $\mathfrak{D}_{(\mathfrak{i},\mathfrak{o})}$ with $(\mathfrak{i},\mathfrak{o}) \in \mathcal{I}$ for some finite set $\mathcal{I} \subsetneq \mathbb{R}^2_{>0}$. We will then show that for all $c \in \{1, \ldots, m\}$, and all executions f with $|f|_A, |f|_B \geq 1$, the value $(\mathcal{V}al_{\vec{x}}(f))_c$ of the c-th constraint after executing f on $\vec{x}$ is

$$(\mathcal{V}al_{\vec{x}}(f))_c = \sum_{(\mathfrak{i},\mathfrak{o})\in\mathcal{I}} \left(\mathfrak{i} \cdot \mathfrak{o}^{\mathcal{U}(f)}\right)^{|f|} \sum_{i\in\mathfrak{D}_{(\mathfrak{i},\mathfrak{o})}} \zeta_{i,\mathbf{A}}^{|f|_A}\, \zeta_{i,\mathbf{B}}^{|f|_B}\, \gamma_{c,i}(\vec{x}). \tag{2}$$

Here, all $\zeta_{i,\mathbf{A}}, \zeta_{i,\mathbf{B}}$ are complex numbers of modulus 1, i.e., $|\zeta_{i,\mathbf{A}}| = |\zeta_{i,\mathbf{B}}| = 1$ for all $i \in \{1, \ldots, n\}$, and the functions $\gamma_{c,i}$ are linear maps $\mathbb{R}^n \to \mathbb{C}$. The maps $\gamma_{c,i}$ and the values $\zeta_{i,\mathbf{A}}, \zeta_{i,\mathbf{B}} \in \mathbb{C}$ only depend on the matrices $\mathbf{C}$, $\mathbf{A}$, and $\mathbf{B}$, but not on the specific input $\vec{x}$. While weaker requirements would suffice to ensure that $(\mathcal{V}al_{\vec{x}}(f))_c$ has some closed form, diagonalizability of $\mathbf{A}$ and $\mathbf{B}$ guarantees that it has the form (2), which is crucial for our procedure. Lemma 21 shows that by a lexicographic comparison of those $(\mathfrak{i},\mathfrak{o}) \in \mathcal{I}$ for which the inner sum in (2) is not 0 for all executions f, one can compute which of the pairs $(\mathfrak{i},\mathfrak{o})$ is the "dominant" one. Here, a pair $(\mathfrak{i},\mathfrak{o}) \in \mathcal{I}$ is considered dominant whenever the value of the first factor $\left(\mathfrak{i} \cdot \mathfrak{o}^{\mathcal{U}(f)}\right)^{|f|}$ of (2) grows the fastest if the execution of f is continued (i.e., if $|f| \to \infty$) and the corresponding inner sum is not 0 for all executions f. The dominant pair depends on the specific input $\vec{x}$ and on whether $\mathcal{U}(f)$ is $\mathfrak{p}$ositive or $\mathfrak{n}$egative, and correspondingly, one has to use different lexicographic comparisons to determine the dominant pair. For $d \in \{\mathfrak{n}, \mathfrak{p}\}$, let $\mathfrak{D}_{d,c,\vec{x}}$ denote the set $\mathfrak{D}_{(\mathfrak{i},\mathfrak{o})}$ where the pair $(\mathfrak{i},\mathfrak{o})$ is dominant for input $\vec{x}$ and $c \in \{1, \ldots, m\}$ (and positive $\mathcal{U}(f)$ if $d = \mathfrak{p}$ or negative $\mathcal{U}(f)$ for $d = \mathfrak{n}$). Then, the sign of the "coefficient" $v(f) = \sum_{i\in\mathfrak{D}_{d,c,\vec{x}}} \zeta_{i,\mathbf{A}}^{|f|_A} \zeta_{i,\mathbf{B}}^{|f|_B} \gamma_{c,i}(\vec{x})$ of the dominant pair eventually determines the sign of $(\mathcal{V}al_{\vec{x}}(f))_c$, provided that $|v(f)|$ is large enough (Lemma 23).

In Sect. 4, we consider the rearrangement

$$v(f) \;=\; \underbrace{\textstyle\sum_{i\in\mathfrak{R}_{d,c,\vec{x}}} \gamma_{c,i}(\vec{x})}_{=R} \;+\; \sum_{i\in\mathfrak{C}_{d,c,\vec{x}}} \zeta_{i,\mathbf{A}}^{|f|_A}\, \zeta_{i,\mathbf{B}}^{|f|_B}\, \gamma_{c,i}(\vec{x}) \tag{3}$$

where $\mathfrak{R}_{d,c,\vec{x}} = \{i \in \mathfrak{D}_{d,c,\vec{x}} \mid \zeta_{i,\mathbf{A}} = \zeta_{i,\mathbf{B}} = 1\}$ and $\mathfrak{C}_{d,c,\vec{x}} = \{i \in \mathfrak{D}_{d,c,\vec{x}} \mid \{\zeta_{i,\mathbf{A}}, \zeta_{i,\mathbf{B}}\} \neq \{1\}\}$.

Lemma 28 then gives a necessary condition for non-termination (and hence a sufficient condition for universal termination): If $\vec{x}$ is an eventually non-terminating input, then there must be a $d \in \{\mathfrak{n}, \mathfrak{p}\}$ such that we have $R > 0$ for all constraints c, with R as in (3).

Sect. 5 turns this necessary condition for non-termination into a sufficient condition. To that end, we define the set W of witnesses for eventual non-termination containing all inputs $\vec{x} \in \mathcal{S}^n$ for which there is some $d \in \{\mathfrak{n}, \mathfrak{p}\}$ such that $R > \sum_{i \in \mathfrak{C}_{d,c,\vec{x}}} |\gamma_{c,i}(\vec{x})|$ holds for all constraints c. Lemma 31 shows that all $\vec{x} \in W$ are eventually non-terminating. While this condition is only sufficient for non-termination, we show in Lemma 32 that if the program is non-terminating, then there is also an input in W. So the considered program is non-terminating iff $W \neq \varnothing$, i.e., the program is universally terminating iff $W = \varnothing$ (Cor. 33).

Finally, our decision procedure for UPAST is presented in Sect. 6. Lemma 34 shows that W is semialgebraic and thus the emptiness problem is decidable over the real algebraic numbers, i.e., if $\mathcal{S} = \mathbb{A}$. Moreover, Lemma 36 shows that W can be represented as a finite union of convex semialgebraic sets. Hence, emptiness of W can also be decided over the rationals and integers [20]. Cor. 39 shows how witnesses for non-termination can be obtained from eventually non-terminating inputs $\vec{x} \in W$. We discuss our implementation in the tool SiRop in Sect. 7. For all proofs, we refer to [26]. Moreover, [26] also contains proof sketches for our main results to help understanding the essential proof ideas.

2 Programs & Termination

As usual, let $\overline{\mathbb{Q}} \subseteq \mathbb{C}$ denote the set of algebraic numbers, i.e., the set of all roots of (univariate) polynomials from $\mathbb{Q}[x]$. As mentioned, $\mathbb{A} = \overline{\mathbb{Q}} \cap \mathbb{R}$ denotes the set of algebraic reals, and $[n]$ denotes the set of positive natural numbers below and including n for every $n \in \mathbb{N}$ with $\mathbb{N} = \mathbb{N}_{>0} \cup \{0\}$, i.e., $[n] = \{1, \ldots, n-1, n\}$ for $n \geq 1$ and $[0] = \varnothing$.

We now define our class of simple randomized loops. The program variables range over a semiring $\mathcal{S} \subseteq \mathbb{A}$ with a guard consisting of a conjunction of m strict linear inequations over the program variables, represented by a matrix $\mathbf{C} \in \mathbb{A}^{m \times n}$, and two commuting linear updates $\mathbf{A}, \mathbf{B} \in \mathcal{S}^{n \times n}$ that are diagonalizable (over $\mathbb{C}$). In each loop iteration, the applied update is chosen among $\mathbf{A}, \mathbf{B}$ according to the outcome of a (possibly biased) coin toss.

It is well known (e.g., [17, Thm. 1.3.12]) that two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{n \times n}$ for $n \in \mathbb{N}$ are commuting and diagonalizable iff they are simultaneously diagonalizable, i.e., there is a regular matrix $\mathbf{S} \in \mathbb{C}^{n \times n}$ such that $\mathbf{A} = \mathbf{S}\mathbf{A}_\mathbf{D}\mathbf{S}^{-1}$ and $\mathbf{B} = \mathbf{S}\mathbf{B}_\mathbf{D}\mathbf{S}^{-1}$, where $\mathbf{A}_\mathbf{D} = \operatorname{diag}(a_1, \ldots, a_n) \in \mathbb{C}^{n \times n}$ and $\mathbf{B}_\mathbf{D} = \operatorname{diag}(b_1, \ldots, b_n) \in \mathbb{C}^{n \times n}$ are complex diagonal matrices, and $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ are the eigenvalues of $\mathbf{A}$ and $\mathbf{B}$, respectively. Moreover, if $\mathbf{A}, \mathbf{B} \in \overline{\mathbb{Q}}^{n \times n}$ are algebraic, then $\mathbf{S}, \mathbf{A}_\mathbf{D}, \mathbf{B}_\mathbf{D} \in \overline{\mathbb{Q}}^{n \times n}$ can also be chosen to be algebraic.

▶ **Definition 1** (Simple Randomized Loops). *Let $m, n \in \mathbb{N}_{>0}$, $\mathbf{C} \in \mathbb{A}^{m \times n}$, $p \in [0,1] \cap \mathbb{A}$, and $\mathbf{A}, \mathbf{B} \in \mathcal{S}^{n \times n}$ such that $\mathbf{A}$ and $\mathbf{B}$ are simultaneously diagonalizable.*[3] *Then,*

$$\texttt{while } \mathbf{C}\vec{x} > \vec{0} : \vec{x} \leftarrow \mathbf{A}\vec{x} \oplus_p \mathbf{B}\vec{x}$$

is called a simple randomized loop *(over $\mathcal{S}$) of dimension n with m constraints. In the remainder, we will omit "simple randomized" and just refer to these programs as "loops".*

The meaning of "$\vec{x} \leftarrow \mathbf{A}\vec{x} \oplus_p \mathbf{B}\vec{x}$" is that $\vec{x}$ is updated to $\mathbf{A}\vec{x}$ with probability p and to $\mathbf{B}\vec{x}$ with probability $1-p$. The comparison $\mathbf{C}\vec{x} > \vec{0}$ is understood componentwise, i.e., $\mathbf{C}\vec{x} > \vec{0}$ iff $(\mathbf{C}\vec{x})_c > 0$ for all $c \in [m]$, where $(\mathbf{C}\vec{x})_c$ is the c-th entry of the vector $\mathbf{C}\vec{x}$. To

[3] In the following, we assume $\mathbb{N} \subseteq \mathcal{S}$ and exclude the trivial semiring $\mathcal{S} = \{0\}$ as every simple randomized loop terminates for the input $\vec{0}$.

simplify the notation, from now on we will consider a fixed loop $\mathcal{P}$ of dimension n with m constraints. Moreover, w.l.o.g. we assume $p \in (0,1)$, i.e., $p \notin \{0,1\}$, as otherwise one can set $\mathbf{A}$ to $\mathbf{B}$ if $p = 0$ and $\mathbf{B}$ to $\mathbf{A}$ if $p = 1$, and then replace p by an arbitrary number from $(0,1)$.

A *run* $\mathfrak{r}$ of the loop $\mathcal{P}$ is an infinite word $\mathfrak{r} = \mathfrak{r}_1\mathfrak{r}_2 \ldots \in \{A,B\}^\omega$, where Runs denotes the set of all runs. For a run $\mathfrak{r} = \mathfrak{r}_1 \ldots$, the value $\mathfrak{r}_i \in \{A,B\}$ indicates which of the updates $\mathbf{A}$ or $\mathbf{B}$ was used in the i-th iteration of the loop. Here, A and B are distinct markings even when $\mathbf{A} = \mathbf{B}$. To simplify the notation, we introduce random variables $\mathfrak{r}_i \colon \mathsf{Runs} \to \{A,B\}$ for $i \in \mathbb{N}_{>0}$ that map a run to its i-th element. Note that all such random variables $\mathfrak{r}_i$ are independent and identically distributed. A *(finite) execution* $f \in \bigcup_{k\in\mathbb{N}} \{A,B\}^k$ is a (possibly empty) prefix of a run. Let Path denote the (countable) set of all such finite executions. Given a finite execution $f = \mathfrak{r}_1 \ldots \mathfrak{r}_k$ with $\mathfrak{r}_i \in \{A,B\}$, let $|f| = k$ denote its length. Furthermore, for $s \in \{A,B\}$, $|f|_s = |\{i \in [k] \mid \mathfrak{r}_i = s\}|$ denotes the number of performed updates with update matrix $\mathbf{A}$ or $\mathbf{B}$, respectively, during the execution of f.

Since the definition of runs is independent from the specific input of the loop, the semantics of the loop $\mathcal{P}$ depend only on the value of $p \in (0,1)$. To obtain a probability measure $\mathbb{P}$ for $\mathcal{P}$, one first considers cylinder sets $\mathrm{Pre}_f = \{f\mathfrak{r}_1\mathfrak{r}_2 \ldots \in \mathsf{Runs} \mid \mathfrak{r}_i \in \{A,B\} \text{ for } i \geq 1\}$ for all $f \in \mathsf{Path}$, i.e., Pre_f contains all runs with prefix f. By requiring $\mathbb{P}(\mathrm{Pre}_f) = p^{|f|_A} \cdot (1-p)^{|f|_B}$ for all $f \in \mathsf{Path}$, one obtains a (unique) probability measure $\mathbb{P} \colon \mathcal{F} \to (0,1)$ on the σ-field $\mathcal{F}$ generated by all cylinder sets Pre_f, see, e.g., [2, Thm. 2.7.2].

▶ **Definition 2** (Semantics of Loops). *The* semantics *of $\mathcal{P}$ is given as a probability space* $(\mathsf{Runs}, \mathcal{F}, \mathbb{P})$ *where* $\mathcal{F} = \sigma(\{\mathrm{Pre}_f \mid f \in \mathsf{Path}\})$ *and* $\mathbb{P}(\mathrm{Pre}_f) = p^{|f|_A} \cdot (1-p)^{|f|_B}$.

To capture the behavior of $\mathcal{P}$ on some specific input $\vec{x} \in \mathcal{S}^n$, we introduce a function $\mathcal{V}al_{\vec{x}} \colon \mathsf{Path} \to \mathbb{A}^m$ that associates finite executions $f \in \mathsf{Path}$ with the values of $\mathcal{P}$'s guard $\mathbf{C}\vec{x}$ *after* the execution of f. Recall that $\mathbf{A}$ and $\mathbf{B}$ commute. Hence, we define

$$\mathcal{V}al_{\vec{x}}(\mathfrak{r}_1 \ldots \mathfrak{r}_k) \;=\; \mathbf{C}\left(\prod_{i=1}^{k} \begin{cases} \mathbf{A} & \text{if } f_i = A \\ \mathbf{B} & \text{otherwise} \end{cases}\right)\vec{x} \;=\; \mathbf{C} \cdot \mathbf{A}^{|\mathfrak{r}_1\ldots\mathfrak{r}_k|_A} \cdot \mathbf{B}^{|\mathfrak{r}_1\ldots\mathfrak{r}_k|_B} \cdot \vec{x}$$

where for every matrix $\mathbf{M} \in \mathbb{C}^{n\times n}$, $\mathbf{M}^0$ is the n-dimensional identity matrix.

▶ **Lemma 3** (Values of Constraints). *For any* $f \in \mathsf{Path}$ *and* $\vec{x} = (x_1, \ldots, x_n) \in \mathbb{A}^n$ *we have*

$$\mathcal{V}al_{\vec{x}}(f) = \mathbf{C} \cdot \mathbf{S} \cdot \mathrm{diag}\left(a_1^{|f|_A} \cdot b_1^{|f|_B}, \ldots, a_n^{|f|_A} \cdot b_n^{|f|_B}\right) \cdot \mathbf{S}^{-1} \cdot \vec{x}.$$

▶ **Example 4.** Consider the loop "$\texttt{while}\ \mathbf{C}\vec{x} > \vec{0} \colon \vec{x} \leftarrow \mathbf{A}\vec{x} \oplus_p \mathbf{B}\vec{x}$" with

$$\mathbf{C} = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \qquad \mathbf{A} = \begin{pmatrix} 11 & 5 & -8 \\ 9 & 15 & 8 \\ 7 & -1 & 6 \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} -7 & 5 & 16 \\ 17 & 5 & -16 \\ -9 & 7 & -12 \end{pmatrix}$$

$$\mathbf{S} = \begin{pmatrix} -\mathrm{i} & \mathrm{i} & 3 \\ \mathrm{i} & -\mathrm{i} & 7 \\ 1 & 1 & 1 \end{pmatrix} \qquad \mathbf{S}^{-1} = \begin{pmatrix} \frac{-1+7\mathrm{i}}{20} & \frac{-1-3\mathrm{i}}{20} & \frac{1}{2} \\ \frac{-1-7\mathrm{i}}{20} & \frac{-1+3\mathrm{i}}{20} & \frac{1}{2} \\ \frac{1}{10} & \frac{1}{10} & 0 \end{pmatrix}$$

Then, for $\vec{x} \in \mathbb{A}^3$ and all $f \in \mathsf{Path}$, $\mathcal{V}al_{\vec{x}}(f)$ equals

$$\mathbf{C} \cdot \mathbf{S} \cdot \mathrm{diag}\left((6-8\mathrm{i})^{|f|_A} \cdot (-12+16\mathrm{i})^{|f|_B},\ (6+8\mathrm{i})^{|f|_A} \cdot (-12-16\mathrm{i})^{|f|_B},\ 20^{|f|_A} \cdot 10^{|f|_B}\right) \cdot \mathbf{S}^{-1} \cdot \vec{x}.$$

The value of the c-th constraint after the execution of $f \in \mathsf{Path}$ on the initial value $\vec{x} \in \mathbb{A}^n$ is given by the expression

$$\begin{aligned}
(\mathcal{V}al_{\vec{x}}(f))_c &= (\mathbf{C} \cdot \mathbf{A}^{|f|_A} \cdot \mathbf{B}^{|f|_B} \cdot \vec{x})_c \\
&= (\mathbf{C} \cdot \mathbf{S} \cdot \mathbf{A_D}^{|f|_A} \cdot \mathbf{B_D}^{|f|_B} \cdot \mathbf{S}^{-1} \cdot \vec{x})_c && \text{(Lemma 3)} \\
&= \textstyle\sum_{i\in[n]} (\mathbf{C} \cdot \mathbf{S})_{c,i} \cdot a_i^{|f|_A} \cdot b_i^{|f|_B} \cdot (\mathbf{S}^{-1} \cdot \vec{x})_i \\
&= \textstyle\sum_{i\in[n]} a_i^{|f|_A} \cdot b_i^{|f|_B} \cdot \gamma_{c,i}(\vec{x}) && (4)
\end{aligned}$$

for linear maps $\gamma_{c,i} \colon \mathbb{A}^n \to \mathbb{C}$ with

$$\gamma_{c,i}(\vec{x}) = (\mathbf{C} \cdot \mathbf{S})_{c,i} \cdot (\mathbf{S}^{-1} \cdot \vec{x})_i. \qquad (5)$$

Here, $(\mathbf{C} \cdot \mathbf{S})_{c,i}$ denotes the entry of $\mathbf{C} \cdot \mathbf{S}$ at row c and column i. Moreover, since $\mathbf{C} \in \mathbb{A}^{m\times n}$, $\mathbf{A}, \mathbf{B} \in \mathbb{A}^{n\times n}$, and $\vec{x} \in \mathbb{A}^n$, we have $\gamma_{c,i}(\vec{x}) \in \overline{\mathbb{Q}}$ for all $(c, i) \in [m] \times [n]$.[4] In the following, we refer to the addends $a_i^{|f|_A} \cdot b_i^{|f|_B} \cdot \gamma_{c,i}(\vec{x})$ of the sum in (4) as *constraint terms.*

▶ **Example 5.** Reconsider Ex. 4. Then, $(\mathcal{V}al_{\vec{x}}(f))_1 = \sum_{i\in[3]} a_i^{|f|_A} \cdot b_i^{|f|_B} \cdot \gamma_{1,i}(\vec{x})$ with eigenvalues $a_1 = 6 - 8\mathrm{i}$, $a_2 = 6 + 8\mathrm{i}$, $a_3 = 20, b_1 = -12 + 16\mathrm{i}$, $b_2 = -12 - 16\mathrm{i}$, $b_3 = 10$, and

$$\begin{aligned}
\gamma_{1,1}(\vec{x}) &= \left(-\tfrac{1}{20} + \tfrac{7\mathrm{i}}{20}\right) x_1 - \left(\tfrac{1}{20} + \tfrac{3\mathrm{i}}{20}\right) x_2 + \tfrac{1}{2} x_3 \\
\gamma_{1,2}(\vec{x}) &= \left(-\tfrac{1}{20} - \tfrac{7\mathrm{i}}{20}\right) x_1 - \left(\tfrac{1}{20} - \tfrac{3\mathrm{i}}{20}\right) x_2 + \tfrac{1}{2} x_3 \\
\gamma_{1,3}(\vec{x}) &= \tfrac{11}{10} x_1 + \tfrac{11}{10} x_2.
\end{aligned}$$

▶ **Corollary 6.** *For all $(c, i) \in [m] \times [n]$ and all $\vec{x}, \vec{y} \in \mathbb{A}^n$, due to the definition of $\gamma_{c,i}$ we have $\gamma_{c,i}(\mathbf{A} \cdot \vec{x} + \vec{y}) = a_i \cdot \gamma_{c,i}(\vec{x}) + \gamma_{c,i}(\vec{y})$ and $\gamma_{c,i}(\mathbf{B} \cdot \vec{x} + \vec{y}) = b_i \cdot \gamma_{c,i}(\vec{x}) + \gamma_{c,i}(\vec{y})$.*

The following lemma shows that if one has two pairs of eigenvalues (a, b) and $(\overline{a}, \overline{b})$ where $\overline{a}$ and $\overline{b}$ are the complex conjugates of a and b, then the sum of all linear maps $\gamma_{c,i}(\vec{x})$ where $(a_i, b_i) = (a, b)$ is the complex conjugate of the sum of all linear maps $\gamma_{c,i}(\vec{x})$ where $(a_i, b_i) = (\overline{a}, \overline{b})$. For instance in Ex. 5, (a_1, b_1) are the complex conjugates of (a_2, b_2) and indeed, we have $\gamma_{1,2}(\vec{x}) = \overline{\gamma_{1,1}(\vec{x})}$. This lemma will later be needed to show that when representing $(\mathcal{V}al_{\vec{x}}(f))_c$ and summing up the coefficients of its addends in a suitable way, all resulting coefficients are real numbers (see Remark 18).

▶ **Lemma 7** (Sums of Conjugated Constraint Terms are Real-Valued)**.** *Let $c \in [m]$, let $a, b \in \mathbb{C}$, and let $\gamma_{c,i}$ be the linear map from (5). Then, for all inputs $\vec{x} \in \mathbb{A}^n$ we have $\gamma_2 = \overline{\gamma_1}$ where*

$$\gamma_1 = \textstyle\sum_{i\,\in\,[n],\ (a_i,b_i)\,=\,(a,b)}\ \gamma_{c,i}(\vec{x}) \quad \textit{and} \quad \gamma_2 = \textstyle\sum_{i\,\in\,[n],\ (a_i,b_i)\,=\,(\overline{a},\overline{b})}\ \gamma_{c,i}(\vec{x}).$$

In order to define the notion of termination for $\mathcal{P}$, we first introduce the concept of a run's length by counting the number of iterations until the guard is violated for the first time. Throughout the paper, we use the convention $\min \varnothing = \infty$.

▶ **Definition 8** (Length Of Runs)**.** *For any $\vec{x} \in \mathbb{A}^n$, we define the random variable $\mathcal{L}_{\vec{x}} \colon \mathsf{Runs} \to \mathbb{N} \cup \{\infty\}$ as $\mathcal{L}_{\vec{x}}(\mathfrak{r}_1\mathfrak{r}_2 \ldots) = \min\{k \in \mathbb{N} \mid \mathcal{V}al_{\vec{x}}(\mathfrak{r}_1 \ldots \mathfrak{r}_k) \not> \vec{0}\}$.*

We now define the *expected runtime* of $\mathcal{P}$ for the input $\vec{x}$ as the expectation $\mathbb{E}(\mathcal{L}_{\vec{x}})$. If $\mathbb{E}(\mathcal{L}_{\vec{x}}) = \infty$, we call the corresponding input $\vec{x}$ *non-terminating.* So we consider *positive almost sure termination* [6, 34], where termination corresponds to a finite expected runtime.

[4] This observation will be needed in the final SMT encoding for our decision procedure (see Lemma 34), as we have to encode the coefficients $\gamma_{c,i}(\vec{x})$ for a given $\vec{x} \in \mathbb{A}^n$.

▶ **Definition 9** (Non-Terminating Inputs). *The set of* non-terminating *inputs is* $\mathrm{NT} = \{\vec{x} \in \mathbb{A}^n \mid \mathbb{E}(\mathcal{L}_{\vec{x}}) = \infty\}$.

Consequently, we call $\mathcal{P}$ *terminating* whenever $\mathrm{NT} = \varnothing$ and non-terminating otherwise.

As in [7, 15, 16, 18, 31, 35], we focus on *eventual non-termination* instead of actual non-termination as this allows us to ignore a finite number of initial updates of the loop. In our setting, an input $\vec{x}$ is *eventually non-terminating* if a non-terminating input $\vec{y}$ can be reached by repeated application of the updates in the loop body to $\vec{x}$.

▶ **Definition 10** (Eventual Non-Termination). *We define the set of* eventually non-terminating *inputs as* $\mathrm{ENT} = \bigcup_{j,k \in \mathbb{N}} \{\vec{x} \in \mathbb{A}^n \mid \mathbf{A}^j \mathbf{B}^k \vec{x} \in \mathrm{NT}\}$.

The motivation behind considering ENT instead of NT is that it allows us to "jump" over the first iterations of the loop (where the loop guard might be violated). In this way, we can focus only on the longterm behavior of the loop on a given input.

▶ **Example 11** (Difference Between NT & ENT). Consider the loop "`while` $\mathbf{C}\vec{x} > \vec{0}: \vec{x} \leftarrow \mathbf{A}\vec{x} \oplus_{0.5} \mathbf{B}\vec{x}$" with $\mathbf{C} = \begin{pmatrix} 1 & 1 \end{pmatrix}$ and $\mathbf{A} = \mathbf{B} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$. Then, $\vec{x} = \begin{pmatrix} 1 \\ -2 \end{pmatrix} \notin \mathrm{NT}$ as $\mathbf{C}\vec{x} = -1 \not> 0$, i.e., $\vec{x}$ violates the loop guard. While $\mathbf{A}\vec{x}$ also violates the loop guard (since $\mathbf{C}\mathbf{A}\vec{x} = 0 \not> 0$), we have $\mathbf{C}\mathbf{A}^j\vec{x} > 0$ for all $j \geq 2$. Thus, $\mathbf{A}^2\vec{x} \in \mathrm{NT}$ and therefore, $\vec{x} \in \mathrm{ENT}$.

Considering ENT instead of NT is justified by the fact that $\mathrm{NT} = \varnothing$ iff $\mathrm{ENT} = \varnothing$, as shown by Lemma 12.

▶ **Lemma 12** (Correspondence of ENT & NT). *For any semiring $\mathcal{S} \subseteq \mathbb{A}$, we have $\mathrm{NT} \cap \mathcal{S}^n = \varnothing$ iff $\mathrm{ENT} \cap \mathcal{S}^n = \varnothing$.*

3 On Constraint Terms

In this section we consider the value $\mathcal{V}al_{\vec{x}}(f)$ for $|f| \to \infty$, motivated by our interest in eventual non-termination. In the first part of this section, we represent $(\mathcal{V}al_{\vec{x}}(f))_c$, for $c \in [m]$, as a sum over so-called "constraint term groups", expressed using a quantity $\mathcal{U}(f)$ corresponding to "how much f has deviated from the expected execution". The section's second part then shows that for specific $f \in \mathsf{Path}$ it suffices to only consider certain addends of this sum in order to decide whether $(\mathcal{V}al_{\vec{x}}(f))_c > 0$ as $|f| \to \infty$. This observation will lead to a necessary condition for eventual non-termination in Sect. 4, which will subsequently be turned into a sufficient criterion in Sect. 5.

When executing the loop $\mathcal{P}$, the relative number of times that update $\mathbf{A}$ is selected over $\mathbf{B}$ will intuitively tend towards p with increasing number of iterations. We now consider this relative quantity and additionally subtract p to center its distribution around 0.

▶ **Definition 13** (Deviation From Equilibrium). *The mapping $\mathcal{U}\colon \mathsf{Path} \to [-p, 1-p]$ is defined as $\mathcal{U}(f) = \frac{|f|_A}{|f|} - p$ for every non-empty $f \in \mathsf{Path}$ and $\mathcal{U}(f) = 0$ otherwise.*

We will investigate which addends determine the sign of $(\mathcal{V}al_{\vec{x}}(f))_c$ for $|f| \to \infty$. To this end, we want to express the value of the constraint terms after some finite execution f in terms of $\mathcal{U}(f)$ and $|f|$. The advantage is that for sufficiently long paths f, we know how $|f|$ and $\mathcal{U}(f)$ "behave" (i.e., $|f|$ "tends towards" ∞ and $\mathcal{U}(f)$ is "expected to tend towards" 0).

▶ **Lemma 14** (Normal Form of Constraint Terms). *Let $a^{|f|_A} b^{|f|_B} \gamma$ be a constraint term with $a, b \neq 0$. We write $a, b \in \mathbb{C}$ in polar form as $|a| \zeta_{\mathbf{A}}$ and $|b| \zeta_{\mathbf{B}}$, respectively, where $\zeta_{\mathbf{A}} = \frac{a}{|a|}, \zeta_{\mathbf{B}} = \frac{b}{|b|} \in \mathbb{C}$ are complex units, i.e., $|\zeta_{\mathbf{A}}| = |\zeta_{\mathbf{B}}| = 1$. Then, for any (non-empty) finite execution $f \in \mathsf{Path}$ with $\mathcal{U}(f) \in (-p, 1-p)$ we have*

$$a^{|f|_A} b^{|f|_B} \gamma = \zeta_{\mathbf{A}}^{|f|_A} \zeta_{\mathbf{B}}^{|f|_B} \left(\tfrac{|a|^p}{|b|^{p-1}} \left(\tfrac{|a|}{|b|} \right)^{\mathcal{U}(f)} \right)^{|f|} \gamma.$$

Note that in Lemma 14 we excluded all constraint terms with $a = 0$ or $b = 0$ in order to avoid a division by zero. However, we additionally required[5] $\mathcal{U}(f) \notin \{-p, 1-p\}$ implying $0 < |f|_A$ and $0 < |f|_B$, since $|f| > 0$. Hence, for all constraint terms $a^{|f|_A} b^{|f|_B} \gamma$ with $a = 0$ or $b = 0$ and all considered $f \in \mathsf{Path}$, we have $a^{|f|_A} b^{|f|_B} = 0$ and thus the value of such constraint terms can safely be ignored when computing $(\mathcal{V}al_{\vec{x}}(f))_c$. This leads to the equation

$$\begin{aligned} (\mathcal{V}al_{\vec{x}}(f))_c &= \textstyle\sum_{i \in [n]} a_i^{|f|_A} b_i^{|f|_B} \gamma_{c,i}(\vec{x}) \\ &= \textstyle\sum_{i \in [n],\, a_i, b_i \neq 0} \zeta_{i,\mathbf{A}}^{|f|_A} \zeta_{i,\mathbf{B}}^{|f|_B} \left(\tfrac{|a_i|^p}{|b_i|^{p-1}} \left(\tfrac{|a_i|}{|b_i|} \right)^{\mathcal{U}(f)} \right)^{|f|} \gamma_{c,i}(\vec{x}), \end{aligned} \tag{6}$$

where $\zeta_{i,\mathbf{A}} = \frac{a_i}{|a_i|}$ and $\zeta_{i,\mathbf{B}} = \frac{b_i}{|b_i|}$.

▶ **Example 15.** Transforming the sum from Ex. 5 into the form (6) and setting $p = \frac{1}{2}$ yields

$$\begin{aligned} (\mathcal{V}al_{\vec{x}}(f))_1 = {} & \zeta_{1,\mathbf{A}}^{|f|_A} \zeta_{1,\mathbf{B}}^{|f|_B} \left(10\sqrt{2} \cdot (\tfrac{1}{2})^{\mathcal{U}(f)}\right)^{|f|} \gamma_{1,1}(\vec{x}) \;+\; \zeta_{2,\mathbf{A}}^{|f|_A} \zeta_{2,\mathbf{B}}^{|f|_B} \left(10\sqrt{2} \cdot (\tfrac{1}{2})^{\mathcal{U}(f)}\right)^{|f|} \gamma_{1,2}(\vec{x}) \\ & +\; \zeta_{3,\mathbf{A}}^{|f|_A} \zeta_{3,\mathbf{B}}^{|f|_B} \left(10\sqrt{2} \cdot 2^{\mathcal{U}(f)}\right)^{|f|} \gamma_{1,3}(\vec{x}) \end{aligned}$$

with $\zeta_{1,\mathbf{A}} = e^{-\mathrm{i} \arctan(4/3)}$, $\zeta_{2,\mathbf{A}} = e^{\mathrm{i} \arctan 4/3}$, $\zeta_{3,\mathbf{A}} = 1$ and $\zeta_{1,\mathbf{B}} = e^{\mathrm{i}(\pi - \arctan(4/3))}$, $\zeta_{2,\mathbf{B}} = e^{\mathrm{i}(\arctan(4/3) - \pi)}$, $\zeta_{3,\mathbf{B}} = 1$ for all $f \in \mathsf{Path}$.

By inspecting the right-hand side of (6) it becomes clear that the subexpressions $\frac{|a_i|^p}{|b_i|^{p-1}}$ and $\frac{|a_i|}{|b_i|}$ govern the overall asymptotic growth of $(\mathcal{V}al_{\vec{x}}(f))_c$ as $|f|$ increases. In the following, we group all constraint terms into so-called *constraint term groups* which are sets of indices i corresponding to constraint terms where $\frac{|a_i|^p}{|b_i|^{p-1}}$ and $\frac{|a_i|}{|b_i|}$ have common values $\mathfrak{i}$ and $\mathfrak{o}$. Here, $\mathfrak{i} = \frac{|a_i|^p}{|b_i|^{p-1}}$ is the expression that is important in (6) if $\mathcal{U}(f)$ is <u>i</u>n a region close to 0.[6] If one is <u>o</u>utside such a region, then $\mathfrak{o} = \frac{|a_i|}{|b_i|}$ is important as well.

▶ **Definition 16** (Constraint Term Groups). *For any $(\mathfrak{i}, \mathfrak{o}) \in \mathbb{R}^2_{>0}$, let*

$$\mathfrak{D}_{(\mathfrak{i},\mathfrak{o})} = \{ i \in [n] \mid 0 \notin \{a_i, b_i\},\ \tfrac{|a_i|^p}{|b_i|^{p-1}} = \mathfrak{i},\ \tfrac{|a_i|}{|b_i|} = \mathfrak{o} \}.$$

Moreover, we define the finite *set $\mathcal{I}$ of all pairs $(\mathfrak{i}, \mathfrak{o}) \in \mathbb{R}^2_{>0}$ with $\mathfrak{D}_{(\mathfrak{i},\mathfrak{o})} \neq \varnothing$. For $c \in [m]$ and $\vec{x} \in \mathbb{A}^n$, let $\mathfrak{D}_{(\mathfrak{i},\mathfrak{o}),c,\vec{x}} = \varnothing$ whenever $\sum_{i \in \mathfrak{D}_{(\mathfrak{i},\mathfrak{o})}} \zeta_{i,\mathbf{A}}^{|f|_A} \zeta_{i,\mathbf{B}}^{|f|_B} \gamma_{c,i}(\vec{x}) = 0$ holds for all $f \in \mathsf{Path}$.*[7] *Otherwise let $\mathfrak{D}_{(\mathfrak{i},\mathfrak{o}),c,\vec{x}} = \mathfrak{D}_{(\mathfrak{i},\mathfrak{o})}$. We refer to the sets $\mathfrak{D}_{(\mathfrak{i},\mathfrak{o}),c,\vec{x}}$ as* constraint term groups.

---

[5] The set of runs $\mathfrak{r}$ where $\mathcal{U}(f) \in \{-p, 1-p\}$ for every prefix f of $\mathfrak{r}$ has probability 0, as $\mathcal{U}(f) \in \{p, 1-p\}$ means that only $\mathbf{A}$ or $\mathbf{B}$ has been selected in f. However, we had required $0 < p < 1$.

[6] Note that $a_i^{|f|_A} b_i^{|f|_B}$ is the $|f|$-th power of the weighted geometric mean $a_i^p b_i^{1-p}$ whenever $\mathcal{U}(f) = 0$.

[7] We will show how to check this in Lemma 24. Note that this is not implied by $\sum_{i \in \mathfrak{D}_{(\mathfrak{i},\mathfrak{o})}} \gamma_{c,i}(\vec{x}) = 0$. As a counterexample, consider $\gamma_1 = -1$, $\gamma_2 = 1$ (and thus, $\gamma_1 + \gamma_2 = 0$), and $\zeta_{1,\mathbf{A}} = -1$, $\zeta_{1,\mathbf{B}} = \zeta_{2,\mathbf{A}} = \zeta_{2,\mathbf{B}} = 1$ (and hence, $\zeta_{1,\mathbf{A}} \zeta_{1,\mathbf{B}} \gamma_1 + \zeta_{2,\mathbf{A}} \zeta_{2,\mathbf{B}} \gamma_2 = 2$).

For $c \in [m]$ and non-empty $f \in \mathsf{Path}$ with $\mathcal{U}(f) \in (-p, 1-p)$, (6) can be rearranged to

$$(\mathcal{V}al_{\vec{x}}(f))_c = \sum_{(\mathfrak{i},\mathfrak{o})\in\mathcal{I}} \left(\mathfrak{i}\cdot\mathfrak{o}^{\mathcal{U}(f)}\right)^{|f|} \sum_{i\in\mathfrak{D}_{(\mathfrak{i},\mathfrak{o}),c,\vec{x}}} \zeta_{i,\mathbf{A}}^{|f|_A}\,\zeta_{i,\mathbf{B}}^{|f|_B}\,\gamma_{c,i}(\vec{x}) \qquad (7)$$

with Def. 16.

Lemma 17 shows that for all $a_1, b_1, a_2, b_2 \neq 0$ we have $\frac{|a_1|}{|b_1|} = \frac{|a_2|}{|b_2|}$ and $\frac{|a_1|^p}{|b_1|^{p-1}} = \frac{|a_2|^p}{|b_2|^{p-1}}$ iff $|a_1| = |a_2|$ and $|b_1| = |b_2|$. Thus, if $i, i' \in \mathfrak{D}_{(\mathfrak{i},\mathfrak{o})}$, then $|a_i| = |a_{i'}|$ and $|b_i| = |b_{i'}|$. We will first return to this result in Sect. 4, where we will use that for all $i, i' \in \mathfrak{D}_{(\mathfrak{i},\mathfrak{o})}$ with $a_i, b_i, a_{i'}, b_{i'} \in \mathbb{R}_{>0}$ we have $(a_i, b_i) = (a'_i, b'_i)$. Later on, we will revisit it in Sect. 5.

▶ **Lemma 17** (Equality of Eigenvalues). *Let $a_1, b_1, a_2, b_2 \in \mathbb{R}_{>0}$ be positive reals such that $\frac{a_1}{b_1} = \frac{a_2}{b_2}$ and $\frac{a_1^p}{b_1^{p-1}} = \frac{a_2^p}{b_2^{p-1}}$. Then we have $(a_1, b_1) = (a_2, b_2)$.*

Due to (7), we have to consider sums $\sum_{i\in\mathfrak{D}_{(\mathfrak{i},\mathfrak{o}),c,\vec{x}}} \zeta_{i,\mathbf{A}}^{|f|_A}\zeta_{i,\mathbf{B}}^{|f|_B}\gamma_{c,i}(\vec{x})$ for $\vec{x} \in \mathbb{A}^n$ and $c \in [m]$. While $\zeta_{i,\mathbf{A}}$, $\zeta_{i,\mathbf{B}}$, and $\gamma_{c,i}(\vec{x})$ are complex numbers in general, such sums are always real-valued. The corresponding Remark 18 is an immediate consequence of Lemma 7. Later, this remark will allow us to make statements about the signs of such sums, see Lemma 23.

▶ Remark 18 (Coefficients of Constraint Term Groups are Real-Valued). Let $c \in [m]$ and $(\mathfrak{i},\mathfrak{o}) \in \mathcal{I}$. Then, for all $f \in \mathsf{Path}$ we have $\sum_{i\in\mathfrak{D}_{(\mathfrak{i},\mathfrak{o}),c,\vec{x}}} \zeta_{i,\mathbf{A}}^{|f|_A}\zeta_{i,\mathbf{B}}^{|f|_B}\gamma_{c,i}(\vec{x}) \in \mathbb{A}$.

▶ **Example 19.** We continue Ex. 15. There are two different non-empty sets $\mathfrak{D}_{(\mathfrak{i},\mathfrak{o})}$, i.e., $\mathfrak{D}_{(10\sqrt{2},1/2)} = \{1,2\}$ and $\mathfrak{D}_{(10\sqrt{2},2)} = \{3\}$. As indicated by Lemma 17, this implies $|a_1| = |a_2|$ and $|b_1| = |b_2|$. Hence, $\zeta_{2,\mathbf{A}} = \frac{a_2}{|a_2|} = \frac{a_2}{|a_1|} = \frac{\overline{a_1}}{|a_1|} = \overline{\zeta_{1,\mathbf{A}}}$ and similarly, $\zeta_{2,\mathbf{B}} = \overline{\zeta_{1,\mathbf{B}}}$. Moreover, recall that $\gamma_{1,2}(\vec{x}) = \overline{\gamma_{1,1}(\vec{x})}$. Therefore, we have

$$\begin{aligned}
& \mathfrak{D}_{(10\sqrt{2},1/2),1,\vec{x}} \neq \varnothing \\
\iff\ & \text{there is a } f \in \mathsf{Path} \text{ with } \zeta_{1,\mathbf{A}}^{|f|_A}\cdot\zeta_{1,\mathbf{B}}^{|f|_B}\cdot\gamma_{1,1}(\vec{x}) + \zeta_{2,\mathbf{A}}^{|f|_A}\cdot\zeta_{2,\mathbf{B}}^{|f|_B}\cdot\gamma_{1,2}(\vec{x}) \neq 0 \\
\iff\ & \text{there is a } f \in \mathsf{Path} \text{ with } \zeta_{1,\mathbf{A}}^{|f|_A}\cdot\zeta_{1,\mathbf{B}}^{|f|_B}\cdot\gamma_{1,1}(\vec{x}) + \overline{\zeta_{1,\mathbf{A}}}^{|f|_A}\cdot\overline{\zeta_{1,\mathbf{B}}}^{|f|_B}\cdot\overline{\gamma_{1,1}(\vec{x})} \neq 0 \\
\iff\ & \text{there is a } f \in \mathsf{Path} \text{ with } \zeta_{1,\mathbf{A}}^{|f|_A}\cdot\zeta_{1,\mathbf{B}}^{|f|_B}\cdot\gamma_{1,1}(\vec{x}) + \overline{\zeta_{1,\mathbf{A}}^{|f|_A}\cdot\zeta_{1,\mathbf{B}}^{|f|_B}\cdot\gamma_{1,1}(\vec{x})} \neq 0 \\
\iff\ & \text{there is a } f \in \mathsf{Path} \text{ with } \mathrm{Re}(\zeta_{1,\mathbf{A}}^{|f|_A}\cdot\zeta_{1,\mathbf{B}}^{|f|_B}\cdot\gamma_{1,1}(\vec{x})) \neq 0 \\
\iff\ & \text{there is a } f \in \mathsf{Path} \text{ with } \zeta_{1,\mathbf{A}}^{|f|_A}\cdot\zeta_{1,\mathbf{B}}^{|f|_B}\cdot\gamma_{1,1}(\vec{x}) \neq 0 \qquad \text{(by Remark 18)} \\
\iff\ & \gamma_{1,1}(\vec{x}) \neq 0
\end{aligned}$$

For the last step, the direction "$\Longrightarrow$" is clear, since $\gamma_{1,1}(\vec{x}) = 0$ implies $\zeta_{1,\mathbf{A}}^{|f|_A}\cdot\zeta_{1,\mathbf{B}}^{|f|_B}\cdot\gamma_{1,1}(\vec{x}) = 0$ for all $f \in \mathsf{Path}$. The direction "$\Longleftarrow$" is also clear by choosing f to be the empty path.

Hence, $\mathfrak{D}_{(10\sqrt{2},1/2),1,\vec{x}} \neq \varnothing \iff \gamma_{1,1}(\vec{x}) \neq 0 \iff 10x_3 \neq x_1 + x_2 \vee 7x_1 \neq 3x_2$. Similarly, $\mathfrak{D}_{(10\sqrt{2},2),1,\vec{x}} \neq \varnothing \iff \gamma_{1,3}(\vec{x}) \neq 0 \iff x_1 + x_2 \neq 0$ for $\vec{x} \in \mathbb{A}^3$.

By inspecting (7) again, one observes that if $\mathcal{U}(f)$ is sufficiently close to 0, then the terms belonging to the non-empty constraint term group $\mathfrak{D}_{(\mathfrak{i},\mathfrak{o}),c,\vec{x}}$ with maximal $(\mathfrak{i},\mathfrak{o})$ in the lexicographic ordering will at some point (when $|f| \to \infty$) outgrow all other terms whenever $\mathcal{U}(f)\cdot|f| > 0$ is **p**ositive and sufficiently large. If $\mathcal{U}(f)\cdot|f| < 0$ is **n**egative and sufficiently small, then, however, we have to focus our attention on the non-empty group $\mathfrak{D}_{(\mathfrak{i},\mathfrak{o}),c,\vec{x}}$ for which $(\mathfrak{i}, -\mathfrak{o})$ is maximal in the lexicographic ordering.[8]

[8] Since $\mathfrak{i}$ and $\mathfrak{o}$ are always positive reals, the expressions $\sum_{i\in\mathfrak{D}_{(\mathfrak{i},\mathfrak{o}),c,\vec{x}}} \zeta_{i,\mathbf{A}}^{|f|_A}\zeta_{i,\mathbf{B}}^{|f|_B}\gamma_{c,i}(\vec{x})$ determine whether $(\mathcal{V}al_{\vec{x}}(f))_c$ is positive or negative. However, for $|f| \to \infty$, this expression (and also $\mathcal{U}(f)$) could alternate between being positive, negative, or even 0. This will be regarded in Sect. 4 and 5.

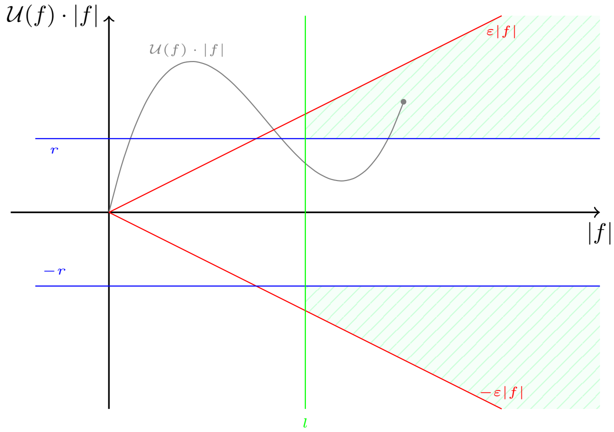

Figure 1 Illustration of the General Idea Using Safe Regions.

▶ **Definition 20** (Eventually Dominating Constraint Term Group). *Let* $\leq_{\text{lex},\mathfrak{p}} = \{((\mathfrak{i}_1,\mathfrak{o}_1),(\mathfrak{i}_2,\mathfrak{o}_2)) \mid \mathfrak{i}_1 < \mathfrak{i}_2$ *or both* $\mathfrak{i}_1 = \mathfrak{i}_2$ *and* $\mathfrak{o}_1 \leq \mathfrak{o}_2\}$ *denote the usual lexicographic ordering on* $\mathbb{R}$ *and let* $\leq_{\text{lex},\mathfrak{n}} = \{((\mathfrak{i}_1,\mathfrak{o}_1),(\mathfrak{i}_2,\mathfrak{o}_2)) \mid \mathfrak{i}_1 < \mathfrak{i}_2$ *or both* $\mathfrak{i}_1 = \mathfrak{i}_2$ *and* $\mathfrak{o}_2 \leq \mathfrak{o}_1\}$ *be the lexicographic ordering where the comparison on the second component is flipped.*

For all $d \in \{\mathfrak{n},\mathfrak{p}\}$, $c \in [m]$, *and* $\vec{x} \in \mathbb{A}^n$ *we define* $\mathfrak{D}_{d,c,\vec{x}} = \varnothing$ *if* $\mathfrak{D}_{(\mathfrak{i},\mathfrak{o}),c,\vec{x}} = \varnothing$ *for all* $(\mathfrak{i},\mathfrak{o}) \in \mathcal{I}$, *and* $\mathfrak{D}_{d,c,\vec{x}} = \mathfrak{D}_{(\mathfrak{i},\mathfrak{o}),c,\vec{x}}$ *for* $(\mathfrak{i},\mathfrak{o}) = \max_{\text{lex},d}\{(\mathfrak{i},\mathfrak{o}) \in \mathcal{I} \mid \mathfrak{D}_{(\mathfrak{i},\mathfrak{o}),c,\vec{x}} \neq \varnothing\}$ *otherwise, where* $\max_{\text{lex},d}$ *denotes the maximum w.r.t. the ordering* $\leq_{\text{lex},d}$.

It is not a priori clear, how, for a constraint index $c \in [m]$ and a given input $\vec{x} \in \mathbb{A}^n$, the sets $\mathfrak{D}_{\mathfrak{n},c,\vec{x}}, \mathfrak{D}_{\mathfrak{p},c,\vec{x}}$ can be computed automatically since one has to decide whether $\frac{a_1^p}{b_1^{p-1}} < \frac{a_2^p}{b_2^{p-1}}$ where $a_1, b_1, a_2, b_2 \in \mathbb{A}_{>0}$ are assumed to be positive algebraic reals, but a_i^p and b_i^{p-1} are in general non-algebraic reals. This is due to the well-known Gelfond-Schneider theorem (see, e.g., [29, Thm. 3.0.1]), which states that $a^p \notin \overline{\mathbb{Q}}$ whenever $a \in \overline{\mathbb{Q}} \setminus \{0,1\}$ and $p \in \overline{\mathbb{Q}}$ is irrational. However, Lemma 21 ensures the decidability of such comparisons.

▶ **Lemma 21** (Comparing Constraint Term Groups). *Let* $T_1 = \frac{a_1^p}{b_1^{p-1}}$ *and* $T_2 = \frac{a_2^p}{b_2^{p-1}}$ *for positive algebraic reals* $a_1, b_1, a_2, b_2 \in \mathbb{A}_{>0}$ *and* $p \in \mathbb{A} \cap (0,1)$. *Then the statement* $T_1 < T_2$ *is decidable.*

▶ **Example 22.** Recall the two non-empty constraint term groups $\mathfrak{D}_{(10\sqrt{2},1/2),1,\vec{x}}$ and $\mathfrak{D}_{(10\sqrt{2},2),1,\vec{x}}$ from Ex. 19. Then, for $\vec{x} \in \mathbb{A}^3$, we have

$$\mathfrak{D}_{\mathfrak{n},1,\vec{x}} = \begin{cases} \mathfrak{D}_{(10\sqrt{2},1/2)} = \{1,2\} & \text{if } 10x_3 \neq x_1 + x_2 \text{ or } 7x_1 \neq 3x_2 \\ \mathfrak{D}_{(10\sqrt{2},2)} \quad = \{3\} & \text{if } 10x_3 = x_1 + x_2,\ 7x_1 = 3x_2, \text{ and } x_1 + x_2 \neq 0 \\ \varnothing & \text{otherwise} \end{cases}$$

$$\mathfrak{D}_{\mathfrak{p},1,\vec{x}} = \begin{cases} \mathfrak{D}_{(10\sqrt{2},2)} \quad = \{3\} & \text{if } x_1 + x_2 \neq 0 \\ \mathfrak{D}_{(10\sqrt{2},1/2)} = \{1,2\} & \text{if } x_1 + x_2 = 0 \text{ and } (10x_3 \neq x_1 + x_2 \text{ or } 7x_1 \neq 3x_2) \\ \varnothing & \text{otherwise} \end{cases}$$

Lemma 23 states the main property of the eventually dominating constraint term groups $\mathfrak{D}_{\mathfrak{n},c,\vec{x}}$ and $\mathfrak{D}_{\mathfrak{p},c,\vec{x}}$. Fig. 1 depicts this lemma (and also the following lemmas) graphically. Here, the horizontal axis represents the length $|f|$ of the path and the vertical axis represents the value of the function $\mathcal{U}(f) \cdot |f| = |f|_A - p \cdot |f|$, which expresses the deviation of the number of A-symbols in the execution f from the expected number of A-symbols. We depicted $\mathcal{U}(f)\cdot|f|$ by a gray line. The lemma essentially states that whenever $\mathcal{U}(f)\cdot|f|$ reaches one of the two "safe" regions marked in green, then the coefficient $v(f)$ of the dominant addend determines the sign of $(\mathcal{V}al_{\vec{x}}(f))_c$, provided that its absolute value $|v(f)|$ is large enough. The upper safe region is the one for $d=\mathfrak{p}$, i.e., here the path f is long enough (i.e., $|f|\geq l$), $\mathcal{U}(f)\in[0,\varepsilon]$ (i.e., $0\leq\mathcal{U}(f)\cdot|f|\leq\varepsilon\cdot|f|$), and $\mathcal{U}(f)\cdot|f|\geq r$. Similarly, the lower safe region corresponds to the case $d=\mathfrak{n}$. This lemma also indicates why an extension of our approach to programs with three instead of two update matrices would be problematic. Then instead of $\mathcal{U}(f)$ we would need a vector to express how much an execution deviates from the probabilities in the program. This would break our concepts of eventually dominating constraint term groups and safe regions, since instead of $d\in\{\mathfrak{n},\mathfrak{p}\}$, we would have to consider the "direction" of this deviation.

▶ **Lemma 23** (Domination of Eventually Dominating Constraint Term Groups)**.** *Let $c\in[m]$, $d\in\{\mathfrak{n},\mathfrak{p}\}$, and $\vec{x}\in\mathbb{A}^n$. We define $v\colon \mathsf{Path}\to\mathbb{A}$ (see Remark 18) as*

$$v(f)=\textstyle\sum_{i\in\mathfrak{D}_{d,c,\vec{x}}}\zeta_{i,\mathbf{A}}^{|f|_A}\zeta_{i,\mathbf{B}}^{|f|_B}\gamma_{c,i}(\vec{x})$$

Then for every $\rho\in\mathbb{A}_{>0}$, there exist constants $\varepsilon\in\mathbb{A}_{>0}$, $r\in\mathbb{N}$, and $l\in\mathbb{N}_{>0}$ (for a bound on the length of the path), such that for all $f\in\mathsf{Path}$ with $|f|\geq l$, $|v(f)|\geq\rho$, $|\mathcal{U}(f)\cdot|f||\geq r$, $\mathcal{U}(f)\in[-\varepsilon,0]$ if $d=\mathfrak{n}$, and $\mathcal{U}(f)\in[0,\varepsilon]$ if $d=\mathfrak{p}$, we have $\operatorname{sign}((\mathcal{V}al_{\vec{x}}(f))_c)=\operatorname{sign}(v(f))$.

4 Positive Eigenvalues

Recall that we are interested in $\operatorname{sign}((\mathcal{V}al_{\vec{x}}(f))_c)$ as the execution progresses, i.e., for $|f|\to\infty$. By Lemma 23, to this end we have to consider the sign of $v(f)=\sum_{i\in\mathfrak{D}_{d,c,\vec{x}}}\zeta_{i,\mathbf{A}}^{|f|_A}\zeta_{i,\mathbf{B}}^{|f|_B}\gamma_{c,i}(\vec{x})$, where $\zeta_{i,\mathbf{A}}=\frac{a_i}{|a_i|}$ and $\zeta_{i,\mathbf{B}}=\frac{b_i}{|b_i|}$. If both a_i and b_i are positive reals, then for $|f|\to\infty$, the sign of $\zeta_{i,\mathbf{A}}^{|f|_A}\zeta_{i,\mathbf{B}}^{|f|_B}\gamma_{c,i}(\vec{x})$ does not change. Thus, we now investigate $\sum_{i\in\mathfrak{D}_{d,c,\vec{x}}}\zeta_{i,\mathbf{A}}^{|f|_A}\zeta_{i,\mathbf{B}}^{|f|_B}\gamma_{c,i}(\vec{x})$ restricted to all $i\in\mathfrak{D}_{d,c,\vec{x}}$ where a_i or b_i is not from $\mathbb{A}_{>0}$. In Lemma 24, we will show that this sum is either always 0 (for all paths f) or it becomes negative for large enough $|f|$.

Assume that for some constraint $c\in[m]$, $d\in\{\mathfrak{n},\mathfrak{p}\}$, and input $\vec{x}$, some eigenvalue of each constraint term in the eventually dominating constraint term group is not positive real, i.e., for all $i\in\mathfrak{D}_{d,c,\vec{x}}$ one has $\zeta_{i,\mathbf{A}}\neq1$ or $\zeta_{i,\mathbf{B}}\neq1$. Then, the sign of the real part of this constraint term will change throughout the program's execution (i.e., for $|f|\to\infty$). Lemma 24 shows that if the sum of these constraint terms is not always 0, then irrespective of the updates that were already performed in previous iterations, this sum becomes smaller than some negative constant C after a number of further iterations. This is expressed in Lemma 24(b), where we have already performed j_0 updates with the matrix $\mathbf{A}$ and k_0 updates with the matrix $\mathbf{B}$. Then by extending the run long enough, the real part of the sum becomes smaller than a constant C that does not depend on j_0 and k_0. Our Lemma 24 is a generalization of a similar result by Braverman [7, Lemma 4] to products of orbits of complex units, i.e., to products of ζ^j for $|\zeta|=1$.[9]

[9] Note that Lemma 24 allows us to check whether $\zeta_{i,\mathbf{A}}^{|f|_A}\zeta_{i,\mathbf{B}}^{|f|_B}\gamma_{c,i}(\vec{x})=0$ holds for all $f\in\mathsf{Path}$, see Footnote 7. By Remark 18, the sum $\sum_{i\in\mathfrak{D}_{(\mathfrak{i},\mathfrak{o})}}\zeta_{i,\mathbf{A}}^{|f|_A}\zeta_{i,\mathbf{B}}^{|f|_B}\gamma_{c,i}(\vec{x})$ is a real number and thus, in the case of (a), $\sum_{i\in\mathfrak{D}_{(\mathfrak{i},\mathfrak{o})}}\zeta_{i,\mathbf{A}}^{|f|_A}\zeta_{i,\mathbf{B}}^{|f|_B}\gamma_{c,i}(\vec{x})=0$ holds for all $f\in\mathsf{Path}$. So given an actual input $\vec{x}$, one just has to check the condition of Lemma 24(a). If that condition does not hold, then by Lemma 24(b), $\sum_{i\in\mathfrak{D}_{(\mathfrak{i},\mathfrak{o})}}\zeta_{i,\mathbf{A}}^{|f|_A}\zeta_{i,\mathbf{B}}^{|f|_B}\gamma_{c,i}(\vec{x})=0$ does not hold for every $f\in\mathsf{Path}$.

▶ **Lemma 24** (Coefficients of Complex Eigenvalues Become Negative). *Let $\gamma_1, \dots, \gamma_l \in \mathbb{C}$ be complex numbers and let $\zeta_{1,1}, \dots, \zeta_{1,l}, \zeta_{2,1}, \dots, \zeta_{2,l} \in \{z \in \mathbb{C} \mid |z| = 1\}$ be complex units such that $\zeta_{1,i} \neq 1$ or $\zeta_{2,i} \neq 1$ for all $i \in \{1, \dots, l\}$. For all $j, k \in \mathbb{N}$, let $z_{j,k} = \sum_{i=1}^{l} \zeta_{1,i}^j \zeta_{2,i}^k \gamma_i$. If all tuples $(\zeta_{1,i}, \zeta_{2,i})$ for $i \in \{1, \dots, l\}$ are pairwise different, then there exist constants $C \in \mathbb{A}_{<0}$ and $K \in \mathbb{N}$ such that we either have (a) or (b):*

(a) *For all $i \in [l]$ with $\gamma_i \neq 0$ there is some $i' \in [l]$ with $\zeta_{1,i} = \overline{\zeta_{1,i'}}$, $\zeta_{2,i} = \overline{\zeta_{2,i'}}$, and $\gamma_i = -\overline{\gamma_{i'}}$, which implies $\mathrm{Re}(z_{j,k}) = \frac{z_{j,k} + \overline{z_{j,k}}}{2} = 0$ for all $j, k \in \mathbb{N}$.*

(b) *For all $j_0, k_0 \in \mathbb{N}$ there exist $j, k \in \{0, \dots, K\}$ such that $\mathrm{Re}(z_{j_0+j,k_0+k}) \leq C$ and there are $j, j', k, k' \in \mathbb{N}$ such that $z_{j,k} \neq z_{j',k'}$.*

▶ **Example 25.** To illustrate Lemma 24, we continue Ex. 22 and consider the constraint term group $\mathfrak{D}_{(10\sqrt{2}, 1/2)} = \{1, 2\}$. Let $\vec{x} \in \mathbb{A}^3$. According to Ex. 15, the coefficient of this constraint term group (for $c = 1$) and $f \in \mathsf{Path}$ is

$$\sum_{i \in \mathfrak{D}_{(10\sqrt{2},1/2)}} \zeta_{i,\mathbf{A}}^{|f|_A} \zeta_{i,\mathbf{B}}^{|f|_B} \gamma_{1,i}(\vec{x}) = (e^{-\mathrm{i} \arctan(4/3)})^{|f|_A} (e^{\mathrm{i}(\pi - \arctan(4/3))})^{|f|_B} \gamma_{1,1}(\vec{x}) + (e^{\mathrm{i} \arctan 4/3})^{|f|_A} (e^{\mathrm{i}(\arctan(4/3) - \pi)})^{|f|_B} \gamma_{1,2}(\vec{x})$$

As already explored in Ex. 19, this coefficient is 0 for all $f \in \mathsf{Path}$ whenever $\gamma_{1,1}(\vec{x}) = 0 \iff \gamma_{1,1}(\vec{x}) = -\overline{\gamma_{1,2}(\vec{x})}$ which corresponds to Lemma 24(a). On the other hand, (b) states that whenever this is not the case, i.e., $\gamma_{1,1}(\vec{x}) \neq 0$, then there are constants $C \in \mathbb{A}_{<0}$, $K \in \mathbb{N}$ such that for every $f \in \mathsf{Path}$ there are $f', f'' \in \mathsf{Path}$ with $|f'|_A, |f'|_B \leq K$ where

$$0 \;>\; C \;\geq\; \sum_{i \in \mathfrak{D}_{(10\sqrt{2},1/2)}} \zeta_{i,\mathbf{A}}^{|ff'|_A} \zeta_{i,\mathbf{B}}^{|ff'|_B} \gamma_{1,i}(\vec{x}) \;\neq\; \sum_{i \in \mathfrak{D}_{(10\sqrt{2},1/2)}} \zeta_{i,\mathbf{A}}^{|ff''|_A} \zeta_{i,\mathbf{B}}^{|ff''|_B} \gamma_{1,i}(\vec{x}),$$

which ends our example to illustrate Lemma 24.

Let $\mathfrak{R} = \{i \in [n] \mid \zeta_{i,\mathbf{A}} = \zeta_{i,\mathbf{B}} = 1\}$ be the set of indices i such that both eigenvalues a_i and b_i are positive reals and let $\mathfrak{C} = [n] \setminus \mathfrak{R} = \{i \in [n] \mid \zeta_{i,\mathbf{A}} \neq 1 \text{ or } \zeta_{i,\mathbf{B}} \neq 1\}$ be the set where at least one of the eigenvalues is not a positive real. To simplify the notation we also denote $\mathfrak{D}_{d,c,\vec{x}} \cap \mathfrak{R}$ and $\mathfrak{D}_{d,c,\vec{x}} \cap \mathfrak{C}$ by $\mathfrak{R}_{d,c,\vec{x}}$ and $\mathfrak{C}_{d,c,\vec{x}}$, respectively, for $(d, c) \in \{\mathfrak{n}, \mathfrak{p}\} \times [m]$. So for all $i \in \mathfrak{R}_{d,c,\vec{x}}$, we have $\zeta_{i,\mathbf{A}}^{|f|_A} \zeta_{i,\mathbf{B}}^{|f|_B} \gamma_{c,i}(\vec{x}) = \gamma_{c,i}(\vec{x})$, i.e., the sign of the corresponding addend does not change for $|f| \to \infty$. For the other eigenvalues, by Lemma 24, $\sum_{i \in \mathfrak{C}_{d,c,\vec{x}}} \zeta_{i,\mathbf{A}}^{|f|_A} \zeta_{i,\mathbf{B}}^{|f|_B} \gamma_{c,i}(\vec{x})$ is either always 0 (for all paths f) or it becomes negative for suitable $|f|_A$ and $|f|_B$.

When executing the loop $\mathcal{P}$ on input $\vec{x}$, one expects that eventually (for $|f| \to \infty$) the constraint term group $\mathfrak{D}_{d,c,\vec{x}}$ for either $d = \mathfrak{n}$ or $d = \mathfrak{p}$ dominates the sign of constraint $c \in [m]$ (Lemma 23). Whenever $\sum_{i \in \mathfrak{R}_{d,c,\vec{x}}} \gamma_{c,i}(\vec{x}) \leq 0$ and $\mathcal{U}(f) \cdot |f|$ has reached one of the two "safe" regions marked in green in Fig. 1, by Lemma 24 one can extend the current path f by a path g_f such that the coefficient $v(f\,g_f) = \sum_{i \in \mathfrak{D}_{d,c,\vec{x}}} \zeta_{i,\mathbf{A}}^{|f\,g_f|_A} \zeta_{i,\mathbf{B}}^{|f\,g_f|_B} \gamma_{c,i}(\vec{x})$ of the dominating constraint term group is negative. Thus, the execution f can be extended by a path g_f such that it leads to termination. This observation is captured in Lemma 26.

▶ **Lemma 26** (Finite Execution leading to Termination). *Let $c \in [m]$, $\vec{x} \in \mathbb{A}^n$, and $d \in \{\mathfrak{n}, \mathfrak{p}\}$, such that $\sum_{i \in \mathfrak{R}_{d,c,\vec{x}}} \gamma_{c,i}(\vec{x}) \leq 0$. Then there are constants $\varepsilon \in \mathbb{A}_{>0}$, $r, u \in \mathbb{N}$, and $l \in \mathbb{N}_{>0}$, such that for all $f \in \mathsf{Path}$ with $|f| \geq l$, $|\mathcal{U}(f) \cdot |f|| \geq r$, $\mathcal{U}(f) \in [-\varepsilon, 0]$ if $d = \mathfrak{n}$, and $\mathcal{U}(f) \in [0, \varepsilon]$ if $d = \mathfrak{p}$, there is a finite execution $g_f \in \mathsf{Path}$ of length $|g_f| \leq u$ with $(\mathcal{V}al_{\vec{x}}(f\,g_f))_c \leq 0$.*

Finally, Lemma 28 builds upon Lemma 26 and gives a sufficient criterion for termination of an input $\vec{x} \in \mathbb{A}^n$. The negation of this criterion is a necessary criterion for every input $\vec{x} \in \mathbb{A}^n$ that is eventually non-terminating. This necessary criterion states that if $\vec{x}$ is eventually non-terminating, then for all constraints c, the sum $\sum_{i \in \mathfrak{R}_{d,c,\vec{x}}} \gamma_{c,i}(\vec{x})$ of the addends for the "dual positive eigenvalues" (where both a_i and b_i are positive reals) must be positive.

So whenever $\sum_{i \in \mathfrak{R}_{d,c,\vec{x}}} \gamma_{c,i}(\vec{x}) \leq 0$, Lemma 28 states that the expected number of steps until $\mathcal{U}(f) \cdot |f|$ reaches a "safe" (green) area in Fig. 1 and executes g_f afterwards is finite. In other words, the expected number of steps $\mathbb{E}(\mathcal{L}_{\vec{x}})$ until termination is finite.

▶ **Example 27.** To motivate Lemma 28 further, we continue Ex. 22. Let $\vec{x} \in \mathbb{A}^3$. We have $\mathfrak{R} = \{3\}$, as only a_3 and b_3 are positive real eigenvalues.

First, suppose $\mathfrak{D}_{\mathfrak{n},1,\vec{x}} \neq \mathfrak{D}_{(10\sqrt{2},2)} = \{3\}$ and $\mathfrak{D}_{\mathfrak{p},1,\vec{x}} \neq \mathfrak{D}_{(10\sqrt{2},2)} = \{3\}$. Then, for all $d \in \{\mathfrak{n}, \mathfrak{p}\}$ we have $\mathfrak{R}_{d,1,\vec{x}} = \varnothing$ and hence $\sum_{i \in \mathfrak{R}_{d,1,\vec{x}}} \gamma_{1,i}(\vec{x}) = 0$. Thus, $\vec{x} \notin \text{ENT}$ by Lemma 28.

On the other hand, if for some $d \in \{\mathfrak{n}, \mathfrak{p}\}$ we have $\mathfrak{D}_{d,1,\vec{x}} = \mathfrak{D}_{(10\sqrt{2},2)} = \{3\}$, i.e., $\mathfrak{R}_{d,1,\vec{x}} = \{3\}$, then Lemma 28 states that $\gamma_{1,3}(\vec{x}) \leq 0$ (i.e., $\frac{11}{10}x_1 + \frac{11}{10}x_2 \leq 0$) implies $\vec{x} \notin \text{ENT}$. However, if $\gamma_{1,3}(\vec{x}) > 0$, then Lemma 28 does not make any statement about whether $\vec{x} \in \text{ENT}$ or $\vec{x} \notin \text{ENT}$.[10]

▶ **Lemma 28** (Dual Positive Eigenvalues for Eventually Dominating Constraints)**.** *Let $\vec{x} \in \mathbb{A}^n$. If for every $d \in \{\mathfrak{n}, \mathfrak{p}\}$ there is a $c \in [m]$ with $\sum_{i \in \mathfrak{R}_{d,c,\vec{x}}} \gamma_i(\vec{x}) \leq 0$, then $\vec{x} \notin$ ENT. Thus, if $\vec{x} \in$ ENT, then there is some $d \in \{\mathfrak{n}, \mathfrak{p}\}$ such that for all $c \in [m]$ we have $\sum_{i \in \mathfrak{R}_{d,c,\vec{x}}} \gamma_{c,i}(\vec{x}) > 0$.*

5 Towards Non-Termination Witnesses

Lemma 28 provides a necessary condition that must hold for all $\vec{x} \in \text{ENT}$. It requires that the sum of the addends $\gamma_{c,i}(\vec{x})$ for all positive real eigenvalues a_i, b_i must be > 0. This condition is however not sufficient for $\vec{x} \in \text{ENT}$. To turn this into a sufficient criterion, we now increase the lower bound 0. More precisely, we replace 0 by the sum of the addends $|\gamma_{c,i}(\vec{x})|$ for all those eigenvalues where a_i or b_i are not a positive real number. In this way, we obtain a sufficient (but no longer necessary) criterion for ENT. To turn this into a sufficient and necessary criterion, we then introduce a "boosting lemma" (Lemma 32), which states that if there is an input in ENT, then there is also a (possibly different) input in ENT that satisfies our sufficient criterion. To prove this boosting lemma, we need the necessary condition of Lemma 28.

For our sufficient (but not necessary) condition for ENT, we define the set of *witnesses* for eventual non-termination as those inputs meeting this criterion.

▶ **Definition 29** (Witnesses for Eventual Non-Termination)**.** *We define the set $W = W_{\mathfrak{n}} \cup W_{\mathfrak{p}}$ of* witnesses for eventual non-termination*, where for $d \in \{\mathfrak{n}, \mathfrak{p}\}$, we have*

$$W_d = \bigcap_{c \in [m]} \left\{ \vec{x} \in \mathbb{A}^n \mid \sum_{i \in \mathfrak{R}_{d,c,\vec{x}}} \gamma_{c,i}(\vec{x}) > \sum_{i \in \mathfrak{C}_{d,c,\vec{x}}} |\gamma_{c,i}(\vec{x})| \right\}.$$

Note that the sum on the left-hand side in the definition of W_d is real-valued due to Lemma 7.

So W_d are all inputs $\vec{x}$ where for all constraints c, the sum of the dominating addends $\gamma_{c,i}(\vec{x})$ for positive real eigenvalues a_i, b_i is greater than the sum of the $|\gamma_{c,i}(\vec{x})|$ for the other eigenvalues a_i, b_i. Lemma 31 shows that the witness condition of Def. 29 is indeed a sufficient criterion for ENT. Before presenting this lemma, we will apply it to our running example.

▶ **Example 30.** We continue Ex. 27. Let $\vec{x} \in \mathbb{A}^3$ with $\gamma_{1,3}(\vec{x}) > 0$ such that for some $d \in \{\mathfrak{n}, \mathfrak{p}\}$ we have $\mathfrak{D}_{d,1,\vec{x}} = \mathfrak{D}_{(10\sqrt{2},2)}$. Then we have $\mathfrak{D}_{d,1,\vec{x}} = \mathfrak{R}_{d,1,\vec{x}} \uplus \mathfrak{C}_{d,1,\vec{x}} = \{3\} \uplus \varnothing$ and thus $\sum_{i \in \mathfrak{R}_{d,1,\vec{x}}} \gamma_{1,i}(\vec{x}) = \gamma_{1,3}(\vec{x}) > 0 = \sum_{i \in \mathfrak{C}_{d,1,\vec{x}}} |\gamma_{1,i}(\vec{x})|$. Hence, $\vec{x} \in W_d \subseteq W$ and thus by the following Lemma 31, we obtain $\vec{x} \in \text{ENT}$, answering the question from Ex. 27.

Hence, $\vec{x} \in \text{ENT}$ iff $\gamma_{1,3}(\vec{x}) > 0$ and there is $d \in \{\mathfrak{n}, \mathfrak{p}\}$ with $\mathfrak{D}_{d,1,\vec{x}} = \mathfrak{D}_{(10\sqrt{2},2)}$. (The "only if" direction is due to Lemma 28, see Ex. 27.)

[10] Sect. 5 will show that in this case one indeed has $\vec{x} \in \text{ENT}$, see Ex. 30.

▶ **Lemma 31** (Witness Criterion is Sufficient for Eventual Non-Termination). *Let $\vec{x} \in \mathbb{A}^n$ be a witness for eventual non-termination, i.e., $\vec{x} \in W$. Then we have $\vec{x} \in \mathrm{ENT}$, i.e., $\vec{x}$ is indeed an eventually non-terminating input.*

Def. 29 introduced a set $W \subseteq \mathbb{A}^n$ that, as shown by Lemma 31, under-approximates the set of eventually non-terminating inputs $\mathrm{ENT} \supseteq W$. While in general we may have $\mathrm{ENT} \supsetneq W$, for the program from Ex. 30 we have $\mathrm{ENT} = W$ as for every $d \in \{\mathfrak{n}, \mathfrak{p}\}$ one either has $\mathfrak{D}_{d,1,\vec{x}} = \mathfrak{R}_{d,1,\vec{x}}$ or $\mathfrak{D}_{d,1,\vec{x}} = \mathfrak{C}_{d,1,\vec{x}}$. So here, $\vec{x} \in \mathrm{ENT}$ implies $\sum_{i \in \mathfrak{R}_{d,1,\vec{x}}} \gamma_{1,i}(\vec{x}) > 0 \Rightarrow \mathfrak{R}_{d,1,\vec{x}} \neq \varnothing \Rightarrow \mathfrak{C}_{d,1,\vec{x}} = \varnothing$ for some $d \in \{\mathfrak{n}, \mathfrak{p}\}$ by Lemma 28 and hence $\sum_{i \in \mathfrak{R}_{d,1,\vec{x}}} \gamma_{1,i}(\vec{x}) > \sum_{i \in \mathfrak{C}_{d,1,\vec{x}}} |\gamma_{1,i}(\vec{x})|$, i.e., $\vec{x} \in W$.

As the set W is rather simple to characterize in contrast to ENT, our goal is to only check for the existence of some $\vec{x} \in W$. This input then witnesses the eventual non-termination of the loop $\mathcal{P}$. The following Lemma 32 establishes that whenever $\mathcal{P}$ is eventually non-terminating, then such a witness $\vec{x} \in W$ does indeed exist. This then leads to our overall decision procedure, because we have that $\mathcal{P}$ is non-terminating $\iff$ $\mathcal{P}$ is eventually non-terminating $\iff$ $W \neq \varnothing$, see Cor. 33. In Sect. 6, we will show that emptiness of W is decidable (not only over the algebraic reals, but also over different sub-semirings $\mathcal{S}$ of $\mathbb{A}$ such as the naturals, integers, or rationals) and if $W \neq \varnothing$, then an element of W is computable.

The intuition behind Lemma 32 is as follows: Given an input $\vec{x} \in \mathrm{NT} \cap \mathcal{S}^n$, we want to construct a non-terminating input in $W \cap \mathcal{S}^n$. Recall that for any constraint $c \in [m]$ and $d \in \{\mathfrak{n}, \mathfrak{p}\}$, the set $\mathfrak{R}_{d,c,\vec{x}}$ contains those indices $i \in \{1, \dots, n\}$ from the dominant constraint term group where the corresponding eigenvalues a_i and b_i of both update matrices $\mathbf{A}$ and $\mathbf{B}$ are positive reals. On the other hand, $\mathfrak{C}_{d,c,\vec{x}}$ contains the remaining indices from the dominant constraint term group. Moreover, the $\gamma_{c,i}(\vec{x})$ help to determine the sign of the corresponding dominant pair's coefficient. If $x \notin W$, then

$$\sum_{i \in \mathfrak{C}_{d,c,\vec{x}}} |\gamma_{c,i}(\vec{x})| \geq \sum_{i \in \mathfrak{R}_{d,c,\vec{x}}} \gamma_{c,i}(\vec{x}) \tag{8}$$

for some $c \in [m]$. One can now modify $\vec{x}$ to make (8)'s left-hand side smaller. For every $i \in \mathfrak{C}_{d,c,\vec{x}}$, if a_i is not positive real, then we multiply $\vec{x}$ by $\mathbf{A}$, and otherwise by $\mathbf{B}$. Since we have $\gamma_{c,i}(\mathbf{A}\vec{x}) = a_i \cdot \gamma_{c,i}(\vec{x})$ by Cor. 6, this "shifts" the phase of at least $\gamma_{c,i}(\vec{x})$ on the left-hand side, but not for the addends on the right-hand side of (8). By performing such multiplications repeatedly and taking a linear combination of the obtained inputs, (8)'s left-hand side becomes arbitrarily small since addends "cancel out", whereas this is not the case for the right-hand side. So one obtains a non-terminating input where (8) does not hold for any $c \in [m]$. Thus, $\vec{x} \in W$.

▶ **Lemma 32** (Boosting). *Let $\mathcal{P}$ be a non-terminating loop over $\mathcal{S}$, i.e., $\mathbf{A}, \mathbf{B} \in \mathcal{S}^{n \times n}$ and $\mathrm{NT} \cap \mathcal{S}^n \neq \varnothing$. Then there is a corresponding witness in $W \cap \mathcal{S}^n$.*

The following corollary summarizes our results so far, i.e., it shows that non-termination is equivalent to the existence of an element in W.

▶ **Corollary 33** (Characterizing Termination). *A loop is terminating over a semiring $\mathcal{S}$ iff $W \cap \mathcal{S}^n = \varnothing$.*

6 Deciding PAST

Finally, we present our novel technique for deciding whether a loop is (positively almost surely) terminating, i.e., whether its expected runtime is finite for every input. As discussed in Sect. 5, to this end we only have to show decidability of $W \neq \varnothing$ for the set of witnesses W for eventual non-termination from Def. 29. We now explain how to translate this emptiness problem into an SMT problem. More precisely, we show that the witness set W is semialgebraic, i.e., it corresponds to a formula over polynomial arithmetic (which is linear in the variables $\vec{x}$). For this we have to take into account that for different values of $\vec{x}$, different addends may be eventually dominating. Then, decidability over the algebraic reals is clear.

As before, $\mathfrak{R}$ are those indices from $[n]$ where both eigenvalues a_i and b_i are positive reals, and $\mathfrak{C}$ are the remaining indices.

▶ **Lemma 34** (Semialgebraic Sets of Witnesses for Algebraic Loops). *Let* $\mathbf{A}, \mathbf{B} \in \mathcal{S}^{n \times n}$, $\mathbf{C} \in \mathbb{A}^{m \times n}$. *We define the sets* $\mathcal{C}^{=0}_{c,(\mathfrak{i},\mathfrak{o})}, \mathcal{C}^{>0}_{c,(\mathfrak{i},\mathfrak{o})} \subseteq \mathbb{A}^n$ *as*

$$\mathcal{C}^{=0}_{c,(\mathfrak{i},\mathfrak{o})} = \left\{ \vec{x} \in \mathbb{A}^n \mid \textstyle\sum_{i \in \mathfrak{D}_{(\mathfrak{i},\mathfrak{o})} \cap \mathfrak{R}} \gamma_{c,i}(\vec{x}) = 0 \right\} \cap \bigcap_{i \in \mathfrak{D}_{(\mathfrak{i},\mathfrak{o})} \cap \mathfrak{C}} H_{i,(\mathfrak{i},\mathfrak{o})}$$

$$H_{i,(\mathfrak{i},\mathfrak{o})} = \left\{ \vec{x} \in \mathbb{A}^n \mid \sum_{\substack{j \in \mathfrak{D}_{(\mathfrak{i},\mathfrak{o})} \\ \zeta_{j,\mathbf{A}} = \zeta_{i,\mathbf{A}} \\ \zeta_{j,\mathbf{B}} = \zeta_{i,\mathbf{B}}}} \gamma_{c,j}(\vec{x}) + \overline{\sum_{\substack{j \in \mathfrak{D}_{(\mathfrak{i},\mathfrak{o})} \\ \zeta_{j,\mathbf{A}} = \overline{\zeta_{i,\mathbf{A}}} \\ \zeta_{j,\mathbf{B}} = \overline{\zeta_{i,\mathbf{B}}}}} \gamma_{c,j}(\vec{x})} = 0 \right\}$$

$$\mathcal{C}^{>0}_{c,(\mathfrak{i},\mathfrak{o})} = \left\{ \vec{x} \in \mathbb{A}^n \mid \textstyle\sum_{i \in \mathfrak{D}_{(\mathfrak{i},\mathfrak{o})} \cap \mathfrak{R}} \gamma_{c,i}(\vec{x}) > \sum_{i \in \mathfrak{D}_{(\mathfrak{i},\mathfrak{o})} \cap \mathfrak{C}} |\gamma_{c,i}(\vec{x})| \right\}$$

for all $(c, (\mathfrak{i}, \mathfrak{o})) \in [m] \times \mathcal{I}$. *Then, for all* $d \in \{\mathfrak{n}, \mathfrak{p}\}$ *and* $\mathfrak{c} = ((\mathfrak{i}_1, \mathfrak{o}_1), \ldots, (\mathfrak{i}_m, \mathfrak{o}_m)) \in \mathcal{I}^m$ *we define* $W_{d,\mathfrak{c}} \subseteq W_d$ *as*

$$W_{d,\mathfrak{c}} = \left\{ \vec{x} \in W_d \;\middle|\; \textstyle\bigwedge_{c \in [m]} \mathfrak{D}_{d,c,\vec{x}} = \mathfrak{D}_{(\mathfrak{i}_c,\mathfrak{o}_c),c,\vec{x}} \right\}.$$

Then

$$W_{d,\mathfrak{c}} = \bigcap_{c \in [m]} \left(\mathcal{C}^{>0}_{c,(\mathfrak{i}_c,\mathfrak{o}_c)} \cap \bigcap_{\substack{(\mathfrak{i}',\mathfrak{o}') \in \mathcal{I} \\ (\mathfrak{i}',\mathfrak{o}') >_{\mathrm{lex},d} (\mathfrak{i}_c,\mathfrak{o}_c)}} \mathcal{C}^{=0}_{c,(\mathfrak{i}',\mathfrak{o}')} \right). \tag{9}$$

Furthermore, we have $W = W_\mathfrak{n} \cup W_\mathfrak{p} = \biguplus_{\mathfrak{c} \in \mathcal{I}^m} W_{\mathfrak{n},\mathfrak{c}} \cup \biguplus_{\mathfrak{c} \in \mathcal{I}^m} W_{\mathfrak{p},\mathfrak{c}}$. *The sets* $W_{d,\mathfrak{c}}$ *and the set* W *are moreover semialgebraic.*

▶ **Example 35.** We continue Ex. 30 and consider $\vec{x} = \begin{pmatrix} 1 & 1 & 0 \end{pmatrix}^{\mathrm{T}} \in \mathbb{N}^3$. By Ex. 30 we have $\vec{x} \in \mathcal{C}^{>0}_{1,(10\sqrt{2},2)}$. Moreover, there is no $(\mathfrak{i}', \mathfrak{o}') \in \mathcal{I} = \{(10\sqrt{2}, 2), (10\sqrt{2}, 1/2)\}$ with $(\mathfrak{i}', \mathfrak{o}') >_{\mathrm{lex},\mathfrak{p}} (10\sqrt{2}, 2)$. For $\mathfrak{c} = (10\sqrt{2}, 2)$ this implies $\vec{x} \in W_{\mathfrak{p},\mathfrak{c}} \subseteq W_\mathfrak{p} \subseteq W \subseteq \mathrm{ENT} \neq \varnothing \Rightarrow \mathrm{NT} \neq \varnothing$.

Thus, the loop initially introduced in Ex. 4 is non-terminating for all $\mathcal{S} \subseteq \{\mathbb{Z}, \mathbb{Q}, \mathbb{A}\}$.[11]

Note that while for $(\mathfrak{i}, \mathfrak{o}) \in \mathcal{I}$ the number $\mathfrak{i}$ is not necessarily algebraic, the representation of $W_d \cap \mathbb{A}^n$ as a finite union/intersection of the semialgebraic sets $\mathcal{C}^{=0}_{(\mathfrak{i},\mathfrak{o}),c}, \mathcal{C}^{>0}_{(\mathfrak{i},\mathfrak{o}),c}$ is still computable by Lemma 21 as one simply has to determine the corresponding ordering $>_{\mathrm{lex},d}$ on $\mathcal{I}$. This is the reason why we restricted p to the set of algebraic reals.

To show that emptiness of W is also decidable over various sub-semirings $\mathcal{S}$ of the algebraic reals, we prove the convexity of the sets $W_{d,\mathfrak{c}}$. Note that the set W as well as the sets $W_\mathfrak{n}, W_\mathfrak{p}$ themselves are in general *not* convex.

[11] We did not consider $\mathcal{S} \in \{\mathbb{N}, \mathbb{Q}_{\geq 0}, \mathbb{A}_{\geq 0}\}$ as for such a choice of $\mathcal{S}$ we do not have $\mathbf{A}, \mathbf{B} \in \mathcal{S}^{n \times n}$.

▶ **Lemma 36** (W_d as Finite Union of Convex Sets). *For $d \in \{\mathfrak{n}, \mathfrak{p}\}$ and $\mathfrak{c} \in \mathcal{I}^m$, the set $W_{d,\mathfrak{c}}$ is convex, i.e., $t\vec{x} + (1-t)\vec{y} \in W_{d,\mathfrak{c}}$ for all $\vec{x}, \vec{y} \in W_{d,\mathfrak{c}}$ and $t \in (0, 1)$.*

Note that Lemmas 34 and 36 imply that for $d \in \{\mathfrak{n}, \mathfrak{p}\}$ the set $W \cap \mathbb{A}^n$ is semialgebraic and a finite union of convex sets.

▶ **Theorem 37** (Deciding PAST). *Let $\mathcal{S} \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}_{\geq 0}, \mathbb{Q}, \mathbb{A}_{\geq 0}, \mathbb{A}\}$. Then, the question whether a loop is terminating on $\mathcal{S}^n$ is decidable, and if the loop is non-terminating, then a witness $\vec{x} \in W \cap \mathcal{S}^n$ for eventual non-termination can be computed.*

▶ Remark 38. The theory of the reals and the algebraic reals are elementary equivalent as both are real closed fields. Thus, Thm. 37 directly entails that the question whether there exists a non-terminating non-negative real input $\vec{x} \in \mathbb{R}^n_{\geq 0}$ or real input $\vec{x} \in \mathbb{R}^n$ for an algebraic loop $\mathcal{P}$ is decidable as well, if one extends the set NT and the corresponding definitions to real inputs $\vec{x} \in \mathbb{R}^n$. Note that in this case $\mathrm{NT} \neq \varnothing$ iff $\mathrm{NT} \cap \mathbb{A}^n \neq \varnothing$.

While the procedure outlined in the proof of Thm. 37 only allows for the computation of a witness $\vec{x} \in \mathrm{ENT} \cap \mathcal{S}^n$ for eventual non-termination, one can lift this to the computation of a witness $\vec{y} \in \mathrm{NT} \cap \mathcal{S}^n$ according to the constructive proofs of Lemmas 23 and 31.

▶ **Corollary 39** (Computing Witnesses for Non-Termination). *Let $\mathcal{S} \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}_{\geq 0}, \mathbb{Q}, \mathbb{A}_{\geq 0}, \mathbb{A}\}$. If a loop is non-terminating, then a witness for non-termination from $\mathrm{NT} \cap \mathcal{S}^n$ can be computed.*

7 Implementation and Conclusion

Prototype Implementation. To demonstrate the practical applicability of our decision procedure, we implemented it in our prototype tool SiRop (for "Simple Randomized Loops"). The tool and a corresponding collection of exemplary programs can be obtained from

`https://github.com/aprove-developers/SiRop`

The tool is implemented in Python and uses the SageMath open-source mathematics software system [33] in order to perform necessary computations such as simultaneous diagonalization and determining the mappings $\gamma_{c,i}$. SiRop tries to compute a witness $\vec{x} \in W$ for eventual non-termination by creating a corresponding SMT problem which is then solved using the SMT-RAT [11] solver. If the SMT problem is unsatisfiable, then the program is terminating. In contrast, if such a witness $\vec{x}$ is found, then the program is non-terminating and the tool computes a non-terminating input $\vec{y} \in \mathrm{NT}$ from $\vec{x}$. Currently, SiRop handles loops over the algebraic reals only, i.e., $\mathcal{S} = \mathbb{A}$, as for all other considered sub-semirings of $\mathbb{A}$, the decision procedure relies on the technique presented in [20] which (to the best of our knowledge) has not yet been implemented.

Conclusion. We have shown the decidability of universal positive almost sure termination (UPAST) for the class of simple randomized loop ranging over numerous semirings $\mathcal{S}$, thereby transferring a line of research started in 2004 by Tiwari [35] on universal termination of linear loops to the realm of randomized programs. To that end, we devised a corresponding decision procedure and presented a prototype implementation for the case $\mathcal{S} = \mathbb{A}$, showing the practical applicability of the presented approach. In particular, our tool managed to find a non-terminating algebraic input for one[12] of the only two problems from the category `C`

[12] `https://github.com/TermCOMP/TPDB/blob/11.3/C_Integer/Stroeder_15/ChenFlurMukhopadhyay-SAS2012-Ex2.06_false-termination.c`

`Integer` which were not solved by any tool at the 2023 Termination Competition [14],[13] the other one being the Collatz problem. While our tool only considers $\mathcal{S} = \mathbb{A}$ (whereas the problem is formulated over the integers), the constraints generated by SiRop are unsatisfiable over $\mathbb{Z}$, which implies universal termination of the program over the integers.

Future Work. While our procedure can decide positive almost sure termination for all inputs, in the future we want to improve it such that it can also compute bounds on expected runtimes. Moreover, decision procedures for termination or complexity of subclasses of non-randomized programs (e.g., [12, 15, 16, 32]) have been integrated in (incomplete) tools that analyze general programs [22, 23], and we would like to investigate such an integration for randomized programs as well. Finally, we plan to adapt our approach to a decision procedure for universal almost sure termination (UAST), i.e., whether a program terminates with probability 1 on all inputs. Clearly, UPAST implies UAST but the converse does not hold in general.

References

1 Sheshansh Agrawal, Krishnendu Chatterjee, and Petr Novotný. Lexicographic ranking supermartingales: An efficient approach to termination of probabilistic programs. *Proc. ACM Program. Lang.*, 2(POPL):34:1–34:32, 2018. doi:10.1145/3158122.

2 Robert B. Ash and Catherine A. Doléans-Dade. *Probability and Measure Theory*. Academic Press, 2nd ed. edition, 2000.

3 Martin Avanzini, Georg Moser, and Michael Schaper. A modular cost analysis for probabilistic programs. *Proc. ACM Program. Lang.*, 4(OOPSLA), 2020. doi:10.1145/3428240.

4 Ezio Bartocci, Laura Kovács, and Miroslav Stankovic. Automatic generation of moment-based invariants for prob-solvable loops. In *Proc. ATVA '19*, LNCS 11781, pages 255–276, 2019. doi:10.1007/978-3-030-31784-3_15.

5 Kevin Batz, Mingshuai Chen, Sebastian Junges, Benjamin Lucien Kaminski, Joost-Pieter Katoen, and Christoph Matheja. Probabilistic program verification via inductive synthesis of inductive invariants. In *Proc. TACAS '23*, LNCS 13994, pages 410–429, 2023. doi:10.1007/978-3-031-30820-8_25.

6 Olivier Bournez and Florent Garnier. Proving positive almost-sure termination. In *Proc. RTA '05*, LNCS 3467, pages 323–337, 2005. doi:10.1007/978-3-540-32033-3_24.

7 Mark Braverman. Termination of integer linear programs. In *Proc. CAV '06*, LNCS 4144, pages 372–385, 2006. doi:10.1007/11817963_34.

8 Tomás Brázdil, Krishnendu Chatterjee, Antonín Kucera, Petr Novotný, and Dominik Velan. Deciding fast termination for probabilistic VASS with nondeterminism. In *Proc. ATVA '19*, LNCS 11781, pages 462–478, 2019. doi:10.1007/978-3-030-31784-3_27.

9 Aleksandar Chakarov and Sriram Sankaranarayanan. Probabilistic program analysis with martingales. In *Proc. CAV '13*, LNCS 8044, pages 511–526, 2013. doi:10.1007/978-3-642-39799-8_34.

10 Krishnendu Chatterjee, Hongfei Fu, and Amir Kafshdar Goharshady. Termination analysis of probabilistic programs through positivstellensatz's. In *Proc. CAV '16*, LNCS 9779, pages 3–22, 2016. doi:10.1007/978-3-319-41528-4_1.

11 Florian Corzilius, Gereon Kremer, Sebastian Junges, Stefan Schupp, and Erika Ábrahám. SMT-RAT: an open source C++ toolbox for strategic and parallel SMT solving. In *Proc. SAT '15*, LNCS 9340, pages 360–368, 2015. doi:10.1007/978-3-319-24318-4_26.

12 Florian Frohn and Jürgen Giesl. Termination of triangular integer loops is decidable. In *Proc. CAV '19*, LNCS 11562, pages 426–444, 2019. doi:10.1007/978-3-030-25543-5_24.

[13] This category was not part of the 2024 competition.

13 Jürgen Giesl, Peter Giesl, and Marcel Hark. Computing expected runtimes for constant probability programs. In *Proc. CADE '19*, LNCS 11716, pages 269–286, 2019. doi:10.1007/978-3-030-29436-6_16.

14 Jürgen Giesl, Albert Rubio, Christian Sternagel, Johannes Waldmann, and Akihisa Yamada. The termination and complexity competition. In *Proc. TACAS '19*, LNCS 11429, pages 156–166, 2019. Website of the Annual Termination Competition: https://termination-portal.org/wiki/Termination_Competition. doi:10.1007/978-3-030-17502-3_10.

15 Marcel Hark, Florian Frohn, and Jürgen Giesl. Polynomial loops: Beyond termination. In *Proc. LPAR '20*, EPiC 73, pages 279–297, 2020. doi:10.29007/nxv1.

16 Marcel Hark, Florian Frohn, and Jürgen Giesl. Termination of triangular polynomial loops. *Formal Methods in Syst. Des.*, 65(1), 2025. doi:10.1007/s10703-023-00440-z.

17 Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2 edition, 2012. doi:10.1017/CBO9781139020411.

18 Mehran Hosseini, Joël Ouaknine, and James Worrell. Termination of linear loops over the integers. In *Proc. ICALP '19*, LIPIcs 132, 2019. doi:10.4230/LIPIcs.ICALP.2019.118.

19 Benjamin Lucien Kaminski, Joost-Pieter Katoen, and Christoph Matheja. On the hardness of analyzing probabilistic programs. *Acta Informatica*, 56(3):255–285, 2019. doi:10.1007/S00236-018-0321-1.

20 Leonid Khachiyan and Lorant Porkolab. Computing integral points in convex semi-algebraic sets. In *Proc. FOCS '97*, pages 162–171, 1997. doi:10.1109/SFCS.1997.646105.

21 Andrey Kofnov, Marcel Moosbrugger, Miroslav Stankovic, Ezio Bartocci, and Efstathia Bura. Exact and approximate moment derivation for probabilistic loops with non-polynomial assignments. *ACM Trans. Model. Comput. Simul.*, 34(3):18:1–18:25, 2024. doi:10.1145/3641545.

22 Nils Lommen and Jürgen Giesl. Targeting completeness: Using closed forms for size bounds of integer programs. In *Proc. FroCoS '23*, LNCS 14279, pages 3–22, 2023. doi:10.1007/978-3-031-43369-6_1.

23 Nils Lommen, Fabian Meyer, and Jürgen Giesl. Automatic complexity analysis of integer programs via triangular weakly non-linear loops. In *Proc. IJCAR '22*, LNCS 13385, pages 734–754, 2022. doi:10.1007/978-3-031-10769-6_43.

24 Éléanore Meyer, Jürgen Giesl, and Sophia Greiwe. SiRop. Software, DFG-Research Training Group 2236 UnRAVeL, swhId: swh:1:dir:3ca664cebef79bfeb95ec944ddc8441d3b528bf6 (visited on 2025-08-05). URL: https://github.com/aprove-developers/SiRop, doi:10.4230/artifacts.24333.

25 Fabian Meyer, Marcel Hark, and Jürgen Giesl. Inferring expected runtimes of probabilistic integer programs using expected sizes. In *Proc. TACAS '21*, LNCS 12651, pages 250–269, 2021. doi:10.1007/978-3-030-72016-2_14.

26 Éléanore Meyer and Jürgen Giesl. Deciding termination of simple randomized loops, 2025. doi:10.48550/arXiv.2506.18541.

27 Marcel Moosbrugger, Ezio Bartocci, Joost-Pieter Katoen, and Laura Kovács. The probabilistic termination tool Amber. *Formal Methods Syst. Des.*, 61(1):90–109, 2022. doi:10.1007/S10703-023-00424-Z.

28 Marcel Moosbrugger, Miroslav Stankovic, Ezio Bartocci, and Laura Kovács. This is the moment for probabilistic loops. *Proc. ACM Program. Lang.*, 6(OOPSLA2):1497–1525, 2022. doi:10.1145/3563341.

29 Saradha Natarajan and Ravindranathan Thangadurai. *Pillars of Transcendental Number Theory*. Springer, 2020. doi:10.1007/978-981-15-4155-1.

30 Van Chan Ngo, Quentin Carbonneaux, and Jan Hoffmann. Bounded expectations: Resource analysis for probabilistic programs. In *Proc. PLDI '18*, pages 496–512, 2018. doi:10.1145/3192366.3192394.

31 Joël Ouaknine, João Sousa Pinto, and James Worrell. On termination of integer linear loops. In *Proc. SODA '15*, pages 957–969, 2015. doi:10.1137/1.9781611973730.65.

32 Enric Rodríguez-Carbonell and Deepak Kapur. Automatic generation of polynomial loop invariants: Algebraic foundations. In *Proc. ISSAC '04*, pages 266–273, 2004. `doi:10.1145/1005285.1005324`.

33 SageMath*, the Sage Mathematics Software System (Version 10.3)*, 2024. `https://www.sagemath.org`.

34 Nasser Saheb-Djahromi. Probabilistic LCF. In *Proc. MFCS '78*, LNCS 64, pages 442–451, 1978. `doi:10.1007/3-540-08921-7_92`.

35 Ashish Tiwari. Termination of linear programs. In *Proc. CAV '04*, LNCS 3114, pages 70–82, 2004. `doi:10.1007/978-3-540-27813-9_6`.

36 Di Wang, David M. Kahn, and Jan Hoffmann. Raising expectations: Automating expected cost analysis with types. *Proc. ACM Program. Lang.*, 4(ICFP), 2020. `doi:10.1145/3408992`.