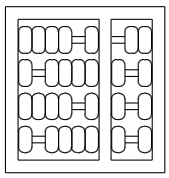

INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

Characterization of Termination for Linear Homogeneous Programs

Rachid Rebiha *Arnaldo V. Moura*
Nadir Matringe

Technical Report - IC-14-08 - Relatório Técnico

June - 2014 - Junho

Characterization of Termination for Linear Homogeneous Programs

Rachid Rebiha* Arnaldo Vieira Moura† Nadir Matringe ‡

Abstract

We present necessary and sufficient conditions for the termination of linear homogeneous programs.We also develop a powerful computational method to determine termination for this class of programs. Our complete characterization of termination for such programs is based on linear algebraic methods. We reduce the verification of the termination problem to checking the orthogonality of a well determined vector space and a certain vector, both related to loops in the program. Moreover, we provide theoretical results and symbolic computational methods guaranteeing the soundness, completeness and numerical stability of the approach. Finally, we show that it is enough to interpret variable values over a specific countable number field, or even over its ring of integers, when one wants to check termination over the reals.

1 Introduction

Static program analysis [1, 2, 3] is used to check that a software is free of defects, such as buffer overflows or segmentation faults, which are safety properties, or termination, which is a liveness property. Verification of temporal properties of infinite state systems [4] is another example. Proving termination of while loop programs is necessary for the verification of liveness properties that any well behaved and engineered system, or any safety critical embedded system, must guarantee. We could list here many verification approaches that are only practical depending on the facility with which termination can be automatically determined. More recent work on automated termination analysis of imperative loop programs has focused on partial

*Instituto de Computação, Universidade Estadual de Campinas, 13081970 Campinas, SP. Pesquisa desenvolvida com suporte financeiro da FAPESP, processos 2011089471 e FAPESP BEPE 2013047349

†Instituto de Computação, Universidade Estadual de Campinas, 13081970 Campinas, SP.

‡Université de Poitiers, Laboratoire Mathématiques et Applications and Institue de Mathematiques de Jussieu Université Paris 7-Denis Diderot, France.

decision procedures based on the discovery and synthesis of ranking functions. Such functions map the loop variable to a well-defined domain where their value decreases at each iteration of the loop [5, 6]. Several interesting approaches, based on the generation of *linear* ranking functions, have been proposed [7, 8] for loop programs where the guards and the instructions can be expressed in a logic supporting linear arithmetic. For the generation of such functions there are effective heuristics [9, 6] and, in some cases, there are also complete methods for the synthesis of such linear ranking functions [10]. On the other hand, it is easy to generate a simple linear terminating loop program that does not have a linear ranking function. In this case those complete synthesis methods [10] fail to provide a conclusion on the termination or nontermination of such a program.

In this work we are motivated by the termination problem for linear while loop programs. In this class of loop programs, the loop condition is a conjunction of linear inequalities and the assignments to each of the variables in the loop instruction block are of affine or linear form. In matrix notation, *linear loop programs* will be represented as: while $(Bx > b)$, $\{x := Ax + c\}$, for x and c in $\mathbb{R}^n$, b in $\mathbb{R}^m$, and A and B real matrices of size $n \times n$ and $m \times n$, respectively. The termination analysis for such class of linear programs can be reduced via different methods, to the termination problem of homogeneous programs with one loop condition, *i.e.* when $m = 1$ and b and c are zero [11, 12], the really difficult point being the reduction to $m = 1$, while the reduction to b and c being zero is immediate. We focus on the termination of this type of program with one loop condition, and obtain results as sharp and complete as one could hope.

At this point, it is worth mentioning our recent work on *asymptotically nonterminant initial variable values* generation techniques [13], where, amongst many other results, we obtain methods that can be adapted here in order to extend our termination analysis for general linear programs, *i.e.* for arbitrary m. Despite tremendous progress over the years [11, 14, 15, 16, 17, 18, 19, 20], the problem of finding a practical, sound and complete method for determining termination or non termination remains very challenging for this class of programs, and for all initial variable values. We started initial investigations following research lines proposed in some preliminary technical reports on termination analysis [21, 22].

We summarize our contributions as follows:

• **Preliminary result**:
First we prove a sufficient condition for the termination of homogeneous linear programs. This result is also stated in the seminal work of [12] but there are several shortcomings in that sketch of the proof leaving space for elaboration. We completed it in a solid mathematical way. We found obstacles which are not easy to fix. We return to this point in more detail at Remark 3.1. Our new proof of this sufficient condition requires nontrivial topological and algebraic arguments. On the other hand, this sufficient condition is not a necessary condition for termination of linear homo-

geneous programs. Before we list our main contributions, it is important to note that the works [12, 11] produces decidability results for our type of programs , however, for programs with one loop condition, our characterisation of termination is simple, very explicit, and gives a straightforward much faster algorithm for termination check. See also Section 7 for a deeper comparaison to Tiwari's and Braverman's methods [12, 11].

• **Main contributions**:
(*i*) We present a *necessary and sufficient condition* (NSC, for short) for the termination of linear homogeneous programs with one loop condition. In fact, this NSC exhibits a complete characterization of termination for such programs, and gives decidability results for all initial values.

(*ii*) Moreover, departing from this NSC, we show the scalability of our approach by demonstrating that one can directly extract a sound and complete computational method to determine termination of such programs. We reduce the termination analysis to the problem of checking if a specific vector, related to the loop encoding condition, belongs to a specific vector space related, to the eigenvalues of the matrix encoding assignments to the loop variables. The analysis of our associated algorithms shows that our method is fast. We show that the proposed computational method, based on three computational steps running in polynomial time complexity, is of a lower complexity than basic routines that form the mathematical foundations of previous methods [12, 11].

(*iii*) We provide theoretical results guaranteeing the soundness and completeness of the termination analysis while restricting variable interpretations over a specific countable sub-ring of $\mathbb{R}^n$. In other words, we show that it is enough to interpret variable values over a specific countable field (a number field, or even over its ring of integers), when one wants to check the termination over the reals. Hereby, we circumvent difficulties such as rounding error. Those results enable our symbolic computational methods to rely on closed-form algebraic expression and numbers.

The rest of this article is organized as follows. Section 2 is as a preliminary section where we introduce our computational model for programs, the notations for the rest of the paper, and some key notions of linear algebra used to develop our computational methods. Section 3 provides our theoretical results and a very useful necessary and sufficient condition, in Subsection 3.2, which allows us to propose the complete computational method illustrated in Section 4 and fully described in Section 5. In the important Section 6, we show that it is enough to interpret the variable values over a countable field in order to determine the program termination over the reals. We provide a discussion of related works in Section 7. Finally, Section 8 concludes the paper.

2 Linear Algebra and Linear Loop Programs

We recall classical facts from linear algebra. Let E be a real vector space and $\mathbf{A}$ belong to $End_{\mathbb{R}}(E)$, the space of $\mathbb{R}$-linear maps from E to itself. We denote by $\mathcal{M}(p,q,\mathbb{R})$ the space of $p \times q$ matrices, and if $p = q$, we simply write $\mathcal{M}(p,\mathbb{R})$. We will denote by $\mathbb{K}$) the field $\mathbb{R}$ or $\mathbb{C}$. If A belongs to $\mathcal{M}(m,n,\mathbb{K})$, with entry $a_{i,j}$ in position (i,j), we will sometimes denote it by $(a_{i,j})$. If B is a basis of E, we denote by $A_B = Mat_B(\mathbf{A})$ the matrix of $\mathbf{A}$ in the basis B, and it belongs to the space $\mathcal{M}(n,\mathbb{R})$. Let I_n the identity matrix in $\mathcal{M}(n,\mathbb{R})$ and $\mathbf{id_E}$ the identity of E. The transpose of the matrix $A = (a_{i,j})$ is the matrix $A^\top = (b_{i,j})$ where $b_{i,j} = a_{j,i}$. The Kernel of A, also called its *nullspace*, denoted by $Ker(A)$, is the set $\{v \in \mathbb{K}^n \mid A \cdot v = 0_{\mathbb{K}^m}\}$. deal with square matrices, these Kernels are *Eigenspaces*. Let A be a square matrix in $\mathcal{M}(n,\mathbb{K})$. A nonzero vector $x \in \mathbb{K}$ is an eigenvector for A associated with the eigenvalue $\lambda \in \mathbb{K}$ if $A \cdot x = \lambda x$, *i.e.*, $(A - \lambda I_n) \cdot x = 0$. The nullspace of $(A - \lambda I_n)$ is called the *eigenspace* of A associated with eigenvalue λ. A non-zero vector x is said to be a *generalized eigenvector* for A corresponding to λ if $(A - \lambda I_n)^k \cdot x = 0$ for some positive integer k. The spaces $Ker((A - \lambda I_n)^k)$ form an increasing sequence of subspaces of E, which is stationary for $k \geq e$, for some $e \leq n$. We call the subspace $Ker((A - \lambda I_n)^e) = Ker((A - \lambda I_n)^n)$ the *generalized eigenspace* of A associated with λ, its nonzero elements are exactly the generalized eigenvectors. We denote by $\langle\ ,\ \rangle$ the canonical scalar product on $\mathbb{R}^n$, and recall, as it is standard in static program analysis, that a primed symbol x' refers to the next state value of x after a transition is taken. Next, we present *transition systems* as representations of imperative programs and *automata* as their computational models.

Definition 2.1. *In a* transition system $\langle x, L, \mathcal{T}, l_0, \Theta\rangle$, *$x = (x_1, ..., x_n)$ is a set of variables, L is a set of locations and $l_0 \in L$ is the initial location. A* state *is given by an interpretation of the variables in x. A* transition *$\tau \in \mathcal{T}$ is given by a tuple $\langle l_{pre}, l_{post}, q_\tau, \rho_\tau\rangle$, where l_{pre} and l_{post} designate the pre- and post-locations of τ, respectively, and the transition relation ρ_τ is a first-order assertion over $x \cup x'$. The transition guard q_τ is a conjunction of inequalities over x. Θ is the initial condition, given as a first-order assertion over x. The transition system is said to be* linear *when ρ_τ is an affine form.*

We will use the following matrix notations to represent loop programs and their transition systems. We also define the class of homogeneous linear programs.

Definition 2.2. *Consider the system $P = \langle x, L, \mathcal{T}, l_0, \Theta\rangle$. where $x = (x_1, ..., x_n)$ and $\mathcal{T} = \langle l, l, q_\tau, \rho_\tau\rangle$. We say that P is a* linear loop program *if:*

- *Transition guards are conjunctions of linear inequalities. We represent the loop condition in matrix form as $Fx > b$ where $F \in \mathcal{M}(m,n,\mathbb{R})$ and $b \in \mathbb{R}^m$. By*

$Fx > b$ we mean that each coordinate of the vector Fx is greater than the corresponding coordinate of vector b.

- *Transition relations are affine or linear forms. We represent the linear assignments in matrix form as $x := Ax + c$, where $A \in \mathcal{M}(n, \mathbb{R})$ and $c \in \mathbb{R}^n$. The most general* linear loop program *$P = P(A, F, b, c)$ is thus written* **while** *$(Fx > b)$ $\{x := Ax + c\}$.*

In this work, one need first to focus manly on the following class of linear loop program.

Definition 2.3. *We denote by $P^{\mathbb{H}}$ the set of programs where all linear assignements consist of* homogeneous *expressions, and where the linear loop condition consists of at most one inequality. If P is in $P^{\mathbb{H}}$, then P will be interpreted in matrix terms as* **while** *$(< f^{\top}, x >> 0)$ $\{x := Ax\}$, where f is a $(n \times 1)$-vector corresponding to the loop condition, and where $A \in \mathcal{M}(n, \mathbb{R})$ is related to the list of assignements in the loop. We say that P has a* homogeneous *form and it will be identified as $P(A, f)$.*

Consider the program $P(A, f)$, where $A \in \mathcal{M}(n, \mathbb{R})$, $f \in \mathcal{M}(1, n, \mathbb{R})$. Alternatively, let $\mathsf{A} \in End_{\mathbb{R}}(E)$, $\mathbf{f} \in E^*$ and $P(\mathsf{A}, \mathbf{f})$: **while** $(\mathbf{f}(\chi) > 0)\{\chi := \mathsf{A}\,\chi\}$. Fixing a basis B of E we can write $A = Mat_B(\mathsf{A})$, $f = Mat_B(\mathbf{f})$, $x = Mat_B(\chi)$, and so on. We now define termination of such a program.

Definition 2.4. *The program $P(\mathsf{A}, \mathbf{f})$ terminates on input $\chi \in E$ if and only if there exists $k \geq 0$ such that $\mathbf{f}(\mathsf{A}^k(\chi))$ is not positive. Also, for $A \in \mathcal{M}_n(\mathbb{R})$, and $f \in \mathcal{M}_{1,n}(\mathbb{R})$, we say that $P(A, f)$ terminates on input $x \in \mathbb{R}^n$, if and only if there exists $k \geq 0$, such that $\langle A^k x, f\rangle$ is not positive.*
Thus, the program $P(\mathsf{A}, \mathbf{f})$ is non-terminating if and only if there exists an input $\chi \in E$ such that $\mathbf{f}(\mathsf{A}^k(\chi)) > 0$ for all $k \geq 0$. In matrix terms, we say that $P(A, f)$ terminates on input vector $x \in \mathbb{R}^n$, if and only if $\langle A^k x, f\rangle > 0$ for all $k \geq 0$.

3 Characterization of Linear Program Termination

First we prove a *sufficient* condition for the termination of homogeneous linear programs, already stated in [12], but with an erroneous proof. Then, we present the main result, which provides the first *necessary and sufficient* condition for the termination problem considering the class of linear homogeneous programs.

3.1 Sufficient Condition for the Termination of Homogeneous Linear Programs

Here, we prove a sufficient condition for the termination of programs $P(A,w) \in P^{\mathbb{H}}$: $\mathsf{while}\ (w^\top x > 0)\ \{x := Ax\}$.

Theorem 3.1. *Let n be a positive integer, and let $P(A,f)$ be a program in $P^{\mathbb{H}}$, If $P(A,f)$ is non-terminating, (i.e. if there exists a vector $x \in \mathbb{R}^n$ such that $\langle A^k x, f\rangle > 0$ for all $k \geq 0$, see Definition 2.4), then A has a positive eigenvalue.*

To complete the proof, which is a mix of topological and algebraic arguments, we need first to sate the following lemmas and propositions. *In the following discussion, we provide the complete proof of Theorem 3.1.* We first recall some basic facts about generalized eigenspaces. Let E be an $\mathbb{R}$-vector space of finite dimension, and let A belong to $End_{\mathbb{R}}(E)$, the space of linear maps from E to itself. Let E' be a subspace of E. We say that E' is A-stable if $\mathsf{A}(E') \subseteq E'$. If $\lambda \in \mathbb{R}$, we denote by $E_\lambda(\mathsf{A})$ the subspace $\{x \in E | \exists k \geq 0, (\mathsf{A} - \lambda \mathbf{id_E})^k(x) = 0\}$. This space is non zero if and only if the input vector x is an eigenvector of A. In this case, it is called a generalized eigenspace. If χ_{A} is the characteristic polynomial of A, if d_λ is the multiplicity of the monomial $(X-\lambda)$ in $\chi_{\mathsf{A}}(X)$, which may be 0 if λ is not an eigenvalue, then $E_\lambda(\mathsf{A}) = Ker(\mathsf{A} - \lambda \mathbf{id_E})^{d_\lambda}$. It is obvious that $E_\lambda(\mathsf{A})$ is A-stable. We denote by $Spec(\mathsf{A})$ the set of real eigenvalues of A. The following property of generalized eigenspaces was stated in the preliminaries.

Proposition 3.1. *Let E be an $\mathbb{R}$-vector space of finite dimension, and let A belong to $End_{\mathbb{R}}(E)$. Then $E_\lambda(\mathsf{A}) = Ker(\mathsf{A} - \lambda \mathbf{id_E})^{d_\lambda}$, for some $d_\lambda \leq n$. In particular, $E_\lambda(\mathsf{A}) = Ker(\mathsf{A} - \lambda \mathbf{id_E})^n$.*

Proof. We just said that one can choose d_λ to be such that $(X-\lambda)^{d_\lambda} \backslash \chi_{\mathsf{A}}$, hence $d_\lambda \leq d^\circ(\chi_{\mathsf{A}}) = n$. □

We will also need the following lemma:

Lemma 3.1. *Let E^* be the space $Hom_{\mathbb{R}}(E,\mathbb{R})$, where E is a finite dimensional vector space, and $f_0,\ldots,f_m$ be linear forms in E^*. Then this family spans E^* if and only if $\cap_{i=0}^m Ker(f_i) = \{0\}$.*

Proof of Lemma 3.1. Suppose that $f_0,\ldots,f_m$ spans E^*, then if x belongs to $\cap_{i=0}^m Ker(f_i)$, then x belongs to the kernel of any element of E^*. But then, if $B = (e_1,\ldots,e_n)$ is a basis of E, and $B^* = (e_1^*,\ldots,e_n^*)$ is its dual basis, we have $x = x_1.e_1 + \cdots + x_n.e_n$, and $e_i^*(x) = x_i = 0$, hence $x = 0$. Conversely, if $\cap_{i=0}^m Ker(f_i) = \{0\}$, Let $g_1,\ldots,g_r$ be a maximal linearly independent family in $f_0,\ldots,f_m$, hence

$$Vect(g_1,\ldots,g_r) = Vect(f_0,\ldots,f_m).$$

We thus have $r \leq n$ (because $dim(E^*) = dim(E) = n$), and $\cap_{i=1}^{r} Ker(g_i) = \{0\}$. If r was strictly smaller than n, then $\cap_{i=1}^{r} Ker(g_i)$ would be an intersection of r subspaces of co-dimension 1, hence it would be of co-dimension at most r, i.e. $\cap_{i=1}^{r} Ker(g_i)$ would be of dimension at least $n - r > 0$, which is absurd, thus $r = n$, and $(g_1, \ldots, g_r)$ is a basis of E^*, thus

$$Vect(f_0, \ldots, f_m) = E^*.$$

□

Before proving Lemma 3.3, we recall and prove a standard Lemma.

Lemma 3.2. *Let A be an endomorphism of a real vector space E, and λ an eigenvalue of A. There is a supplementary space E' of $E_\lambda(\mathsf{A})$ (i.e. $E = E_\lambda(\mathsf{A}) \oplus E'$), and two polynomials C and D in $\mathbb{R}[X]$, such that $C(\mathsf{A})$ is the projection on $E_\lambda(\mathsf{A})$ with respect to E', and $D(\mathsf{A})$ is the projection on E' with respect to $E_\lambda(\mathsf{A})$. In particular E' is also A-stable, and for any A-stable subspace L of E, we have $L = L \cap E_\lambda(\mathsf{A}) \oplus L \cap E'$.*

Proof. Let $\chi_{\mathsf{A}} = (X - \lambda)^d Q$, with $Q(\lambda) \neq 0$. By the Kernel's decomposition Lemma, we have

$$E = Ker(\mathsf{A} - \lambda I_d)^d \oplus Ker(Q(\mathsf{A})).$$

We set $E' = Ker(Q(\mathsf{A}))$. It is thus A-stable. Moreover, by Bezout's identity, there are P and P' in $\mathbb{R}[X]$, such that

$$P(u) \circ (\mathsf{A} - \lambda I_d)^d + P'(u) \circ Q(\mathsf{A}) = I_d,$$

then we set $C = P(X - \lambda)^d$, and $D = P'(\mathsf{A}) \circ Q(u)$. Finally, if L is A-stable, we always have

$$L \cap E_\lambda(\mathsf{A}) \oplus L \cap E' \subset L.$$

Now write an element l of L as $l_1 + l_2$, with $l_1 \in E_\lambda(\mathsf{A})$, and $l_2 \in E'$, we have $B(\mathsf{A})(l) = l_1$, but L being A-stable, it is $D(\mathsf{A})$-stable as well, hence $l_1 \in L$. Similarly we have $l_2 \in L$, thus

$$L = L \cap E_\lambda(\mathsf{A}) \oplus L \cap E'.$$

□

We will use the following result about quotient vector-spaces.

Lemma 3.3. *Let E be an $\mathbb{R}$-vector space, and A belong to $End_{\mathbb{R}}(E)$, and suppose that L is a A-stable subspace of E. Let $\overline{\mathsf{A}} : E/L \to E/L$, be the element of $End_{\mathbb{R}}(E/L)$, defined by $\overline{\mathsf{A}}(x + L) = \overline{\mathsf{A}}(x) + L$. Then $Spec(\overline{\mathsf{A}}) \subset Spec(\mathsf{A})$. More generally, for any $\lambda \in Spec(\overline{\mathsf{A}})$, the generalized eigenspace $E_\lambda(\mathsf{A})$ maps surjectively to $E_\lambda(\overline{\mathsf{A}})$ in E/L.*

Proof of Lemma 3.3. Let B_1 be a basis of L, and B_2 be a basis of any supplementary space. Call $\overline{B_2}$ the image of the elements of B_2 in $\overline{E} = E/L$, then $\overline{B_2}$ is a basis of $\overline{E}$. Let $B = B_1 \cup B_2$, it is a basis of B, and $Mat_B(\mathsf{A})$ is of the form

$$\begin{pmatrix} X & Y \\ 0 & Z \end{pmatrix}.$$

Then $X = Mat_{B_1}(\mathsf{A}_{|L})$, and $Z = Mat_{\overline{B_2}}(\overline{\mathsf{A}})$, and the second statement follows from this second fact.
Now if $\overline{x}$ belongs to $E_\lambda(\overline{\mathsf{A}})$, then

$$(\overline{\mathsf{A}} - \lambda\overline{I_d})^a\overline{x} = \overline{0}$$

for some $a \geq 0$. This means that $(\mathsf{A} - \lambda I_d)^a x \in L$.

We write $x = x_\lambda + x' \in E_\lambda(\mathsf{A}) \oplus E'$, for E' as in Lemma 3.2. Then $(\mathsf{A} - \lambda I_d)^a x = (\mathsf{A} - \lambda I_d)^a x_\lambda + (\mathsf{A} - \lambda I_d)^a x'$, and $(\mathsf{A} - \lambda I_d)^a x_\lambda \in E_\lambda(\mathsf{A})$, and $(\mathsf{A} - \lambda I_d)^a x' \in E'$. Let d be λ's multiplicity as a root of χ_A, for k large enough kd such that $kd \geq a$, we have $(\mathsf{A} - \lambda I_d)^{kd} x_\lambda = 0$ and $(\mathsf{A} - \lambda I_d)^{kd} x = (\mathsf{A} - \lambda I_d)^{kd} x'$. But take $P \in \mathbb{R}[X]$ as in the proof of Lemma 3.2, we obtain that $P(\mathsf{A}) \circ (\mathsf{A} - \lambda I_d)^d$ is the identity when restricted to E', in particular, this implies that

$$x' = P(\mathsf{A})^k (\mathsf{A} - \lambda I_d)^{kd} x,$$

and thus $x' \in L$. Finally, we obtain

$$\overline{x} = \overline{x_\lambda},$$

and this ends the proof as $x_\lambda \in E_\lambda(\mathsf{A})$. □

We say that a subset of $\mathbb{R}^n$ is a convex cone if it is convex, and is stable under multiplication by elements of $\mathbb{R}_{>0}$. It is obvious that an intersection of convex cones is still a convex cone. Hence, one can speak of the convex cone spanned by a subset of $\mathbb{R}^n$.

Proposition 3.2. *Let C be a convex cone of $\mathbb{R}^n$ non reducible to zero, and contained in the closed cone $\Delta = \{x = (x_1, ..., x_n) \in \mathbb{R}^n \,|\, \forall\ i, x_i \geq 0\}$. If A is an invertible endomorphism of $\mathbb{R}^n$, with $A(C) \subset C$, then A has a positive eigenvalue.*

Proof. Consider $C' = C - \{0\}$, then C' is also a convex cone. It is obviously still stable under multiplication by elements of $\mathbb{R}_{>0}$. Moreover, if x and y belong to C', then for $t \in [0, 1]$, the vector $tx + (1 - t)y$ belongs to C by convexity, but it cannot be equal to zero, because otherwise, as both x and y have non negative coefficients, this would imply that x or y is null, which is absurd.
Now let H_1 be the affine hyperplane

$$H_1 = \{x \in \mathbb{R}^n, x_1 + \ldots x_n = 1\},$$

and call f the linear form on $\mathbb{R}^n$, defined by $f : x \mapsto x_1 + \cdots + x_n$, so that $H = f^{-1}(\{1\})$. This linear form is positive on Δ, hence we can define the projection $p : \Delta - \{0\} \to H$, given by

$$x \mapsto \frac{1}{f(x)}x,$$

it is obviously continuous. We call C_1 the set $p(C')$, we claim that $C_1 = C' \cap H_1$, in particular it is convex. Indeed, $C_1 \subset H_1$ by definition, and $C_1 \subset C'$ because C' is stable under $\mathbb{R}_{>0}$. Conversely, the restriction of p to $C' \cap H_1$ is the identity, hence C_1 contains $C' \cap H_1 = p(C' \cap H_1)$. It is also clearly stable under the continuous map

$$s = p \circ A : \Delta - \{0\} \to H_1$$

as $A(C') \subset C'$. In particular, its closure $\overline{C_1}$ is stable under s as well. It is again convex, and compact, as a closed subset of the compact set

$$\{x \in \mathbb{R}^n, \forall\ i,\ x_i \geq 0,\ x_1 + \cdots + x_n = 1\}.$$

According to Brouwer's fixed point theorem, this implies that s has a fixed x point in $\overline{C_1} \subset \Delta - \{0\}$, but we then have $A(x) = f(x)x$. As $f(x) > 0$ for any x in $\Delta - \{0\}$, this proves the Lemma. □

Finally we will prove the following statement equivalent to Theorem 3.1. We just rewrite the statement of Theorem 3.1 in terms of morphisms, since they are more convenient to work with.

Theorem 3.2. *Let E be an $\mathbb{R}$-vector space of dimension n, let A be a endomorphism of E, and f a nonzero linear form on E. If there exists a vector $x \in E$, such that $f(\mathsf{A}^k(x)) > 0$ for all $k \geq 0$, then A has a positive eigenvalue.*

Proof. We prove the result by induction on n. For $n = 1$, we can identify E with $\mathbb{R}$. Then A is of the form $x \mapsto t_{\mathsf{A}}.x$, for some nonzero t_{A}, and $\{f > 0\}$ is either $\mathbb{R}_{>0}$, or $\mathbb{R}_{<0}$. Hence, x belongs to $\mathbb{R}_{>0}$, or $\mathbb{R}_{<0}$, and $t_{\mathsf{A}}^k.x$ belongs to the same half-space for every $k \geq 0$. Hence, $t_{\mathsf{A}} > 0$.

Now if A is non invertible, we can replace E by the image of A, $Im(\mathsf{A})$, and x by $\mathsf{A}(x)$, so that the hypothesis are still verified by A's restriction to $Im(\mathsf{A})$. But $Im(\mathsf{A})$ being a subspace of E of strictly smaller dimension, we get the result using the induction hypothesis. We are thus left with the case A invertible. Let m be the maximal non negative integer such that $(f, f \circ \mathsf{A}, \ldots, f \circ \mathsf{A}^m)$ is a linearly independent family of E^*. It is easy to see that $L = \cap_{k \geq 0} Ker(f \circ \mathsf{A}^k)$ is equal to $\cap_{k=0}^m Ker(f \circ \mathsf{A}^k)$. Hence, it is A-stable. The space L is a proper subspace of E because it is contained in $Ker(f)$. Taking the quotient space $\overline{E} = E/L$, the linear map A induces $\overline{\mathsf{A}} : \overline{E} \to \overline{E}$, and f induces a linear form $\overline{f}$ on $\bar{E}$. By letting $\bar{x}$ be the image of x in E, the quadruplet $(\overline{E}, \overline{\mathsf{A}}, \overline{f}, \bar{x})$ still satisfies the hypothesis of the theorem. If L is not zero,

using the induction we conclude that the linear map $\overline{\mathsf{A}}$ has a positive eigenvalue $\lambda > 0$. But λ is necessarily an eigenvalue of A by Lemma 3.3, and we are done in this case. Finally, assume that $L = \{0\}$. Then $(e_1^* = f, e_2^* = f \circ \mathsf{A}, \ldots, e_n^* = f \circ \mathsf{A}^m)$ is a basis of E^* (in particular $m = n - 1$) according to Lemma 3.1. Take $(e_1, \ldots, e_n)$ as its dual basis in E, and identify E with $\mathbb{R}^n$, given this basis. Then $\mathsf{A}^k(x)$ belongs to the space $\{v \,|\, \forall i, v_i > 0\} \subset \Delta$ for all $k \geq 0$. Hence, the convex cone C spanned by this family as well. It is clearly A-stable, and is not reduced to zero as it contains x. We conclude by applying Proposition 3.2. □

This also concludes the proof of Theorem 3.1, as Theorem 3.2 is an equivalent statement written in terms of morphisms $A = Mat_B(\mathsf{A}))$ and $w^\top = Mat_B(f)$. Theorem 3.1 says that the linear program terminates when there is no positive eigenvalues. But one cannot conclude on the termination problem using theorem 3.1 if there exists at least one positive eigenvalue. As we already mentioned, Theorem 3.1 is stated in [12], but the proof given there contains certain flaws that we now explain.

Remark 3.1. *The author of [12] applies the Brouwer's fixed point theorem to a subspace of the projective space $P(\mathbb{R}^n)$ (and not $\mathbb{R}^{n-1}$ as stated in [12]). However, this is not an euclidian space, and so convexity is not well defined in it. Hence, one cannot apply Brouwer's fixed point theorem to such a set. Moreover, using notation as in the proof of Theorem 1 in [12], the closure NT' of the set NT can contain zero, for example as soon as all, real or complex, eigenvalues of A have their module less than 1. Hence, its image in $P(\mathbb{R}^n)$ is not well defined. The case of NT' containing zero raises a serious problem that needs to be treated carefully, we get rid of it by quotienting by L in our proof.*

Theorem 3.1 provides a sufficient condition for the termination of linear program. In other words, Theorem 3.1 says that the linear program terminates when there is no positive eigenvalues, but one can not conclude on the termination problem using theorem 3.1 if there exists at least one positive eigenvalue. Intuitively, we could say that theorem 3.1 provides us with a decidability result for the termination problem considering the subclass of linear program where the associated assignement matrix A has no positive eigenvalues (i.e., all eigenvalues are complex or negative). In the following example, we illustrate when Theorem 3.1 applies and when it does not.

Example 3.1. *Consider the homogeneous linear program 1a depicted in the figure 1 that we denote by $P(A, v)$. The associated matrix A is given by $A = \begin{pmatrix} 3 & -2 \\ 4 & -1 \end{pmatrix}$, and the vector v enconding the loop condition, is such that $v = (3, -1)^\top$. The eigenvalues of the matrix A are the complex numbers: $1 + 2i$ and $1 - 2i$. As S does not have any positive eigenvalues, we can consider the contrapose of Theorem 3.1's statement, and conclude that the program $P(A, v)$ terminates on all possible inputs.*

```
/*...*/
 while(3x - y > 0){
   x := 3x - 2y;
   y := 4x - y;
   }
/*...*/
```

(a)

```
/*...*/
 while(z > 0){
   x:= x + y;
   z:= -z;
   }
/*...*/
```

(b)

Figure 1: Examples of homogeneous linear programs

Example 3.2. *Now, consider the homogeneous linear program 1b depicted in Figure 1, that we denote by* $P(A_1, v_1)$. *The associated matrix* A_1 *given by* $A_1 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$, *has eigenvalues* 1 *and* -1. *As* A *has a positive eigenvalues, one can not determine the termination (or the nontermination) of* $P(A_1, v_1)$ *using the theorem 3.1.*

However, We will see how to handle this case in a very automated efficient way in the following sections.

In the next section, we generalize Theorem 3.1, and provide a different and stronger decidability results.

3.2 Necessary and Sufficient Condition for the Termination of Linear Programs

In Theorem 3.3 we provide a necessary and sufficient condition for the termination of programs $P(A, w) \in P^{\mathbb{H}}$: $\mathsf{while}\ (w^\top x > 0)\ \{x := Ax\}$.

Theorem 3.3. *Let* $A \in \mathcal{M}_n(\mathbb{R})$ *and* $w \neq 0 \in \mathbb{R}^n$. *The program* $P(A, w)$: $\mathsf{while}\ (w^\top x > 0)\ \{x := Ax\}$ *terminates if and only if for every positive eigenvalue* λ *of* A, *the generalized eigenspace* $E_\lambda(A)$ *is orthogonal to* w, i.e., $w^\top E_\lambda(A) = \langle w, E_\lambda(A)\rangle = 0$.

In order to prove Theorem 3.3 we first restate it in equivalent linear algebraic terms.

Theorem 3.4. *Let* E *be an* $\mathbb{R}$*-vector space of finite dimension* n, *let* A *be an endomorphism of* E, *and let* $\mathbf{f}$ *be a nonzero linear form on* E. *Then, there exists a vector* $x \in E$, *with* $\mathbf{f}(\mathsf{A}^k(x)) > 0$ *for all* $k \geq 0$, *if and only if there is* $\lambda > 0$ *in* $Spec(\mathsf{A})$ *such that* $E_\lambda(\mathsf{A}) \not\subset Ker(\mathbf{f})$.

Proof. First suppose that there is a $\lambda > 0$ in $Spec(\mathsf{A})$ with $E_\lambda(\mathsf{A}) \not\subset Ker(\mathbf{f})$. Then there is some $r \geq 1$ such that $Ker(\mathsf{A} - \lambda\mathbf{id_E})^{r-1} \subset Ker(\mathbf{f})$ but also $Ker(\mathsf{A} - \lambda\mathbf{id_E})^r \not\subset Ker(\mathbf{f})$. Let x be an element of $Ker(\mathsf{A} - \lambda\mathbf{id_E})^r - Ker(\mathbf{f})$ such that $\mathbf{f}(x) > 0$. This

is always possible because $Ker(\mathsf{A}-\lambda\mathbf{id_E})^r - Ker(\mathbf{f})$ is stable under $y \mapsto -y$. Because $x \in Ker(\mathsf{A}-\lambda\mathbf{id_E})^r$, it is clear that $\mathsf{A}(x) - \lambda x \in Ker(\mathsf{A}-\lambda\mathbf{id_E})^{r-1}$. Let L be $Ker(\mathsf{A}-\lambda\mathbf{id_E})^{r-1}$, and $\overline{E} = E/L$. As L is A-stable, $\overline{\mathsf{A}}$ is well defined, and $\overline{\mathsf{A}}(\overline{x}) = \lambda\overline{x}$ because $\mathsf{A}(x) - \lambda x \in L$. Moreover, $L \subset Ker(\mathbf{f})$. Hence, $\overline{\mathbf{f}}$ is well defined and $\overline{\mathbf{f}}(\overline{\mathsf{A}}^k(\overline{x})) = \mathbf{f}(\mathsf{A}^k(x))$ for every $k \geq 0$. As $\overline{\mathsf{A}}^k(\overline{x}) = \lambda^k\overline{x}$, we deduce that $\mathbf{f}(\mathsf{A}^k(x)) = \lambda^k f(x) > 0$ for all $k \geq 0$.
Conversely, suppose that there exists a vector $x \in E$, such that $\mathbf{f}(\mathsf{A}^k(x)) > 0$ for all $k \geq 0$. We are going to prove by induction on n that A has an eigenvalue $\lambda > 0$ such that $E_\lambda(\mathsf{A})$ is not contained in $Ker(\mathbf{f})$. If $n = 1$, then $\mathsf{A} : t \mapsto \lambda t$ for $\lambda \in \mathbb{R}$, and thus, $\lambda^k(\mathbf{f}(x)) > 0$ for all $k \geq 0$, which implies $\lambda > 0$, and $E_\lambda(\mathsf{A}) = E$ which cannot be contained in $Ker(\mathbf{f})$. If $n > 1$, according to Theorem 3.2, we know that A admits a positive eigenvalue μ. If $E_\mu(\mathsf{A})$ is not a subset of $Ker(\mathbf{f})$ we are done. If $L = E_\mu(\mathsf{A}) \subset Ker(\mathbf{f})$, we consider $\overline{E} = E/L$. This vector space is of dimension less than n, and $\overline{\mathbf{f}}(\overline{\mathsf{A}}^k(\overline{x})) = \mathbf{f}(\mathsf{A}^k(x)) > 0$ for all $k \geq 0$. By the induction hypothesis, there is $\lambda > 0$ in $Spec(\overline{\mathsf{A}})$ such that $E_\lambda(\overline{\mathsf{A}}) \not\subset Ker(\overline{\mathbf{f}})$. But λ belongs to $Spec(\mathsf{A})$ according to Lemma 3.3, and $E_\lambda(\mathsf{A})$ maps surjectively on $E_\lambda(\overline{\mathsf{A}})$ according to this same Lemma. In particular, we have $\overline{\mathbf{f}}(E_\lambda(\overline{\mathsf{A}})) = \mathbf{f}(E_\lambda(\mathsf{A}))$, but the left hand side is not reduced to zero in this equality. Hence, $\mathbf{f}(E_\lambda(\mathsf{A})) \neq \{0\}$, *i.e.*, $E_\lambda(\mathsf{A}) \not\subset Ker(\mathbf{f})$, concluding the proof. □

This argument proves Theorem 3.3 as it is a direct corollary of Theorem 3.4 with $A = Mat_B(\mathsf{A}))$ and $f = Mat_B(\mathbf{f})$. Theorem 3.3 gives a necessary and sufficient condition that we can use as the foundation to build a complete procedure for checking termination. In order to determine termination, we have to check, for each positive eigenvalues, if the vector f, encoding the loop condition, is orthogonal to the associated generalized eigenspace. In other words we want to verify if f is orthogonal to the nullspace $Ker((A - \lambda I_n)^n)$.

Example 3.3. *Consider the program 1b depicted in Figure 1 that we denoted as $P(A_1, v_1)$. The matrix A_1 is given in Example 3.1. The vector enconding the loop condition is $v_1 = e_3 = (0, 0, 1)^\top$. We recall that A_1 has eigenvalues 1 and -1. The generalised eigenspace $E_1(A_1)$ is equal to $Vect(e_1, e_2)$, where e_1 and e_2 are the first two vectors of the canonical basis of $\mathbb{R}^3$. Hence $E_1(A_1)$ is orthogonal to v_1. According to Theorem 3.3, the program $P(A, w)$ terminates.*

Example 3.4. *Now, if we change the loop condition of the program 1b depicted in Figure 1 to become $(y > 0)$. Then, we obtain the program $P(A_1, v_2)$ with the new considered loop condition encoded by $v_2 = e_2 = (0, 1, 0)^\top$. The eigenvalues of A_1 are (still) 1 and -1 and the generalised eigenspace $E_1(A_1) = Vect(e_1, e_2)$. Hence $E_1(A)$ is not orthogonal to v_2, because it contains v_2. Theorem 3.3 tells us that the program $P(A_1, v_2)$ does not terminate in this case.*

In both of these examples, we are able to determine the termination/nontermination using Theorem 3.3. On the other hand, the first Theorem 3.1 does not allow us to say anything about the termination of these programs (because the assignement matrix A' exhibit at least one positive eigenvalue). In order to avoid the computation of basis for generalized eigenspaces, we first introduce the space $Row\_Space(M)$, and use the next lemma. If $M \in \mathcal{M}(m, n, \mathbb{R})$, then $Row\_Space(M)$ denotes the vector subspace of $\mathbb{R}^n$ spanned by the row vectors of M.

Lemma 3.4. *Let M be a matrix in $\mathcal{M}(m, n, \mathbb{K})$. Then every vector in the nullspace of M is orthogonal to every vector in $Row\_Space(M)$.*

Proof. Let w be in $Ker(M)$ and v in the column space of $M^\top$. We denote by $\{c_1, ..., c_m\}$ the set of column vectors of $M^\top$. Then, exists a vector $k \in \mathbb{R}^m$ such that $v = M^\top \cdot k$ (because v is a linear combination of the column vectors of $M^\top$). Now, we have $< w, v >= w^\top \cdot v = w^\top \cdot M^\top \cdot k = (M \cdot w)^\top \cdot k = 0$ because $w \in Ker(M)$ and $M \cdot w = 0$. □

From Lemma 3.4, a basis of $Row\_Space(M)$ is a basis of the orthogonals of $Ker(M)$. Thus, for square matrix A, a vector v is orthogonal to $Ker((A - \lambda I_n)^n)$ (i.e. $< E_\lambda(A), v >= 0$) if an only if v belongs to $Row\_Space((A - \lambda I_n)^n)$. We directly deduce the following corollary.

Corollary 3.1. *Let $A \in \mathcal{M}_n(\mathbb{R})$ and $v \neq 0 \in \mathbb{R}^n$. The program $P(A, v)$ terminates if and only if for every positive eigenvalue λ of A, v belongs to the vector space $Row\_Space((A - \lambda I_d)^n)$.*

Proof. By Lemma 3.4, the basis of $Row\_Space((A - \lambda I_d)^n))$ is a basis of the orthogonal of $Ker((A - \lambda I_d)^n))$. We then apply Theorem 3.3. □

4 Running Example

In practice, we can use Corollary 3.1 to support three fast computational steps, as illustrated in the following example.

Example 4.1. (Running example) *Consider the program $P(A, v)$ depicted as follow:*

(i) Pseudo code:

```
while(z+t-x-y>0){
  x := 2x - y;
  y := -x + 2y -z;
  z := -y + 2z +t;
  t := 2t;}
```

(ii) Associated matrices:

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & 1 \\ 0 & 0 & 0 & 2 \end{pmatrix}, \text{ and } v = \begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \end{pmatrix}.$$

Step 1: We compute the list e_λ of positive eigenvalues for A. The result is:

```
[[2 - sqrt(2), sqrt(2) + 2,2], [1, 1, 2]]
```

Hence, we have three positive eigenvalue $\lambda_1 = 2, \lambda_2 = 2 - \sqrt{2}, \lambda_3 = 2 + \sqrt{2}$ *(with, respectively, the multiplicity* 2, 1 *and* 1*).*
Step 2: We compute the matrix $E_\lambda = (A - \lambda I_n)^n$ taking the eigenvalue $\lambda = 2 + \sqrt{2}$. The result is:

```
(A - (e[i])*Id_m)^d
[         18  16*sqrt(2)          14  -4*sqrt(2)]
[ 16*sqrt(2)          32  16*sqrt(2)         -14]
[         14  16*sqrt(2)          18 -12*sqrt(2)]
[          0           0           0           4]
```

Step 3: We check if $v \in Row\_Space(E_\lambda)$:
Here we use a standard procedure, from linear algebra, in order to check if a given vector belongs to a vector-space spanned by a given set of vectors. We compute the unique reduced row echelon form *of the matrix* $E_\lambda^\top$*. For that we run a* Gaussian elimination *on the rows using the* Gauss-Jordan elimination *algorithm. The generated matrix, below on the left, provides us with a linearly independent basis for* $Row\_Space(E_\lambda)$*. We remove the rows containing only zeros entries and we augment the computed basis with the vector* $v^\top$ *by appending it as the last row. Then, we obtain the matrix below on the right.*

```
(E[i]).echelon_form()
[      1       0      -1       0]
[      0       1 sqrt(2)       0]
[      0       0       0       1]
[      0       0       0       0]
```

```
block_matrix([[Er[i]], [V.T]])
[      1       0      -1       0]
[      0       1 sqrt(2)       0]
[      0       0       0       1]
[--------------------------------]
[     -1      -1       1       1]
```

Finally, we generate its reduced row echelon form to obtain the matrix R_S_λ*:*

```
block_matrix([[Er[i]], [V.T]]).echelon_form()
[1 0 0 0]
[0 1 0 0]
[0 0 1 0]
[-------]
[0 0 0 1]
```

From the Gauss-Jordan elimination properties, it is well-known that v *belongs to the space* $Row\_Space(E_\lambda)$ *if and only if* $R\_S_\lambda(n, n+1) = 0$*. Here we have* $R\_S_\lambda(n, n+1) = 1$*, which means that* v *is not in* $Row\_Space(E_\lambda)$*. Thus, by Corollary 3.1, one concludes that the program* $P(A, v)$ *is nonterminant.* □

As we show in Example 4.1, we avoid the computation of generalized eigenspaces in practice. Instead, use the exact algorithm associated to Corollary 3.1.

5 A Complete Procedure to Determine Termination

We use the necessary and sufficient conditions provided by Theorem 3.3 and its related practical Corollary 3.1 to build a sound and complete procedure to establish the termination of linear programs. Moreover, the method so obtained is based on few computational steps associated to fast numerical algorithms. The pseudo code depicted in Algorithm 1 illustrates the strategy. Our algorithm takes as input the number of variables, the chosen field where the variables are interpreted, the assignment matrix A and the vector w encoding the loop condition. We first compute the list of positive eigenvalues (lines 1 and 2 in 1). If this list is empty we can then state that the loop is terminant (lines 3 and 4). Otherwise, we continue the analysis using the nonempty list of positive eigenvalues. For each positive eigenvalues $e'[i]$ we first need to compute the matrix $E_i = (A - e'[i]I_n)^n$ (line 6). Using Corollary 3.1, we know that the loop is terminant if and only if w is in the *Row_Space* of $(A - e'[i]I_n)^n$ for every positive eigenvalue $e'[i]$. In other words, for each positive eigenvalue, we have to check if w is in the vector space spanned by the basis of the *Row_Space* of the associated matrix E_i. In order to do so, one first need to consider the linearly independent vectors $\{r_1, ..., r_n\}$ that form a basis of the *Row_Space*. This basis is obtained from the list of the non-zero row vectors of the computed *reduced row echelon form* of E_i (lines 7 and 8). The efficient way to check if w is in the vector space spanned by the basis $\{r_1, ..., r_n\}$ comprises the following computational steps: (i) We build the augmented matrix E_A formed by the row vectors $r_1, ..., r_n$ and $w^\top$ (line 9); (ii) We compute the *reduced row echelon form* of matrix E_A (line 8). For that we apply *Gaussian elimination* on the rows. This reduced, canonical form is unique and is computed exactly by the *Gauss-Jordan elimination*; (iii) We know that the added vector w is in the vector space spanned by $r_1, ..., r_n$ if and only if the bottom right entry of the reduced row echelon matrix E_R is null. Thus if $E_R(n, n+1) \neq 0$, we conclude that there exists a positive eigenvalue $e'[i]$ such that w is not in $Row\_Space(A - e'[i]I_n)^n$, which is equivalent to saying that the loop is nonterminant (lines 11 and line 12). Otherwise if he have exhausted the list of positive eigenvalues and always found that w is in the *Row_Space* of the associated matrix, we conclude that the loop is terminant (line 13).

The function **echelon_form** computes the reduced row echelon form by Guass-Jordan elimination and its time complexity is of order $O(n^3)$. We interpret the variables in a specified field, *i.e.* an extension of $\mathbb{Q}$, chosen according to the discussion made in section 6. By using efficient mathematical packages, *e.g.* Maple,

Algorithm 1: Termination_linear_Loop $(n, \mathbb{K}, A, w)$

/*Determining the termination for linear homogeneous programs.*/;
Data: n the number of program variables, $\mathbb{K}$ the field, $P(A, w) \in P^{\mathbb{H}}$ where $A \in \mathcal{M}(n, \mathbb{K})$ and $w \in \mathcal{M}(n, 1, \mathbb{K})$
Result: Determine the Termination/Nontermination
begin

```
1  {e[1], ..., e[r]} ⟵ eigenvalues(A);
2  {e'[1], ..., e'[s]} ⟵ striclty_positives({e[1], ..., e[r]});
3  if {e'[1], ..., e'[s]} = ∅ then there is no positive eigenvalues.
4      return TERMINANT;
5  for i = 1 to s do
6      𝔼 ⟵ (A − e'[i]I_n)^n;
7      𝔼_rrf ⟵ echelon_form(𝔼);
8      𝔼'_rrf ⟵ remove_zero_row(𝔼_rrf);
9      𝔼_A ⟵ augment_row(𝔼'_rrf, w^⊤);
10     𝔼_R ⟵ echelon_form(𝔼_A);
11     if 𝔼_R(n, n + 1) ≠ 0 then
12         return NONTERMINANT;
13 return TERMINANT;
```

Mathematica, Sage, Lapack or Eispack, one can expect the eigenvalues to be in closed-form algebraic expressions, *i.e.* the solution of an algebraic equation in terms of the coefficients, relying only on addition, subtraction, multiplication, division, and the extraction of roots. Also, it is well known that with $n < 5$, the eigenvalues computed by the function **eigenvalues** are already exhibited as such algebraic numbers. Moreover, the algorithm for eigenvalue computation has a time complexity that is of order $O(n^3)$ and, this being said, the overall time complexity of the algorithm **Termination_linear_Loop** remains of the same order of time complexity.

In Table 1 we list some experimental results. The column **Set-i** refers to a set of loops generated randomly. The column **#Loops** gives the number of loops treated. We give the associated fields in column **Fi** (e.g., the countable subset described in Section 6). The column **Dim** refers to the dimension of the initial systems, *i.e*, the number of variables. The column **#T** shows the number of programs found to be terminant, and the column **#NT** gives the number of loops programs found to be nontermiant. Finally, column **CPU/s[T]** refers to cpu time results while proving all the terminant loop programs, and column **CPU/s[N]** gives the cpu time taken to establish nontermination of programs. The column **CPU/s[total]** gives the cpu time results in seconds for concluding on the termination for the given set of 500 loops. We have implemented our prototype using **Sage** [23] using interfaces written in Python. By doing so, we were able to have access to several useful mathematical packages. As expected, we can see that more nonterminant programs were found, as they are easier to write. Note also that it takes much more time to prove termination than to prove nontermination.

6 Interpreting the Variables Over Countable Sets is Sufficient

In this section, we show that to analyze the termination of a linear program $P(A, v)$ with one loop conition over $\mathbb{R}^n$, we can restrict our analysis to the case where the variable belongs to a contable subset of $\mathbb{R}^n$ depending on A. First, we study an example, which is already interesting in itself, and which will prove that we cannot restrict the interpretation of the variable to the field $\mathbb{Q}$ of rational numbers if we want to prove the termination for all real inputs. We start with two elements of $\mathbb{Q}(\sqrt{2})-\mathbb{Q}$, which are conjugate under the Galois group $Gal_{\mathbb{Q}}(\mathbb{Q}(\sqrt{2}))$, of opposite signs, and the negative one of absolute value strictly greater than the one with positive absolute value. For instance, take $\lambda^- = -1 - \sqrt{2}$, and $\lambda^+ = -1 + \sqrt{2}$. They are the roots of the polynomial $P(X) = (X-\lambda^-)(X-\lambda^+) = X^2+2X-1$. Now let $A = \begin{pmatrix} 0 & 1 \\ 1 & -2 \end{pmatrix}$ be the associated companion matrix, so that its characteristic polynomial is P, and its eigenvalues are λ^- and λ^+. Its generalized eigenspaces are easy to compute, and we

find: $E_{\lambda^-}(A) = \mathbb{R}.e^-$ and $E_{\lambda^+}(A) = \mathbb{R}.e^+$ with $e^- = \begin{pmatrix} 1 \\ \lambda^- \end{pmatrix}$ and $e^+ = \begin{pmatrix} 1 \\ \lambda^+ \end{pmatrix}$. Now let $v = (1,0)^\top$. We have $< v, e^+ >= 1$ and so, according to Theorem 3.3, the program $P_1 = P(A, v)$, associated to A and v, does not terminate. We can actually locate the points of $\mathbb{R}^2$ for which the program is not terminating.

Proposition 6.1. *Let A, v and P_1 be as above. Then program P_1 does not terminate for an initial condition $x \in \mathbb{R}^2$ if and only if $x \in E_{\lambda^+}(A)$ and $\langle x, v\rangle > 0$, i.e. $x \in \mathbb{R}_{>0}.e^+$.* □

Proof. If $x = t.e^+$, with $t > 0$, then $A^k(x) = t\lambda^{+k}.x$, and $< v, A^k(x) >= t\lambda^{+k} > 0$ for all $k \geq 0$. Hence the program does not terminate for such x as initial condition. Conversely, suppose that x satisfies $\langle v, A^k(x)\rangle > 0$ for all $k \geq 0$. Decompose x on the basis (e^-, e^+). Then $x = s.e^- + t.e^+$, and $A^k(x) = s\lambda^{-k}.e^- + t\lambda^{+k}.e^+$, so that $< v, A^k(x) >= s\lambda^{-k} + t\lambda^{+k}$. Suppose that s is not zero. As $|\lambda^-| > |\lambda^+|$, for k large enough, the scalar $< v, A^k(x) >$ will be of the same sign as $s\lambda^{-k}$, which is alternatively positive and negative. Since this is absurd, $s = 0$. Now as $< v, A^k(x) >= t\lambda^{+k}$, this implies that $t > 0$, and so the Proposition is proved. □ □

Proposition 6.1 leads us to the following corollary.

Corollary 6.1. *With A and v as above, program P_1 is terminating on $\mathbb{Q}^2$, but not on $\mathbb{R}^2$* □

Proof. We already saw that P_1 does not terminate on $\mathbb{R}^2$. Now let x be an element of $\mathbb{Q}^2$. If P_1 was not terminating with x as an initial value, this would imply x in $\mathbb{R}_{>0}.e^+$ according to Lemma 6.1. However, no element of $\mathbb{Q}^2$ belongs to $\mathbb{R}_{>0}.e^+$, because the quotient of the coordinates of e^+ is irrational. This implies that P_1 terminates on $\mathbb{Q}^2$. □ □

This proves that even if A and v are rational, one cannot guarantee the termination over the reals if the interpretation of the variables are restricted to rationals. It is clear that one cannot hope to produce any valid conjecture of this type if A and v have wild coefficients, like transcendentals, for example. However, when A and v have algebraic coefficients, using Corollary 3.1, one can find a simple remedy. It is indeed enough to replace $\mathbb{Q}$ by a finite extension of the field $\mathbb{Q}$. Such an extension K is called a *number field*, and is known to be countable. Indeed, it is a $\mathbb{Q}$-vector space of finite dimension, *i.e.*, $K = \mathbb{Q}.k_1 \oplus \cdots \oplus \mathbb{Q}.k_l$ for some $l \geq 1$, and elements k_i in K. It is, moreover, known that K is the fraction field of its *ring of integers* O_K, which is a free $\mathbb{Z}$-module of finite type. In fact $O_K = \mathbb{Z}.o_1 \oplus \cdots \oplus \mathbb{Z}.o_l$ for the same $l \geq 1$, and where the elements o_i can be chosen equal to the k_i, for well chosen k_i's. We say that a number field is *real* if it is a subfield of $\mathbb{R}$. Notice that in the mathematical literature, a totally real number field is a number field with only real embeddings in $\mathbb{C}$. Here what we call real is thus weaker than totally real.

Theorem 6.1. *Let $A \in \mathcal{M}_n(\mathbb{R})$, $v \neq 0 \in \mathbb{R}^n$, and suppose that their coefficients are actually in $\mathbb{Q}$ or, more generally, in a* real *number field K. Then there is a well-determined* real *finite extension L of $\mathbb{Q}$, or of K in the general case, which is contained in $\mathbb{R}$ and such that the program $P(A, v)$ associated to A and v terminates if and only if it terminates on the countable set L^n. We can choose L to be the extension $\mathbb{Q}(\lambda_1, \ldots, \lambda_t)$ of $\mathbb{Q}$, or $K(\lambda_1, \ldots, \lambda_t)$ in general, spanned by the positive eigenvalues $(\lambda_1, \ldots, \lambda_t)$ of A. It is actually enough to check the termination of the program on O_L^n.* □

Proof. We deal with the general case. The reader not familiar with field extensions can just replace K by $\mathbb{Q}$. It is obvious that if the program terminates, it terminates on L^n for any subset L of $\mathbb{R}$. Now let $\lambda_1, \ldots, \lambda_r$ be the positive eigenvalues of A. They are all roots of the minimal (or characteristic) polynomial Q of A, which belongs to $K[X]$. Hence they are all algebraic on K, and so also on $\mathbb{Q}$ as $K/\mathbb{Q}$ is finite. Let $L = K(\lambda_1, \ldots, \lambda_r) \subset \mathbb{R}$. Suppose that the program P_1 does not terminate. Then there is some $i \in \{1, \ldots, r\}$, such that $< E_{\lambda_i}, v > \neq 0$ according to Corollary 3.3. Let r be the positive integer such that $Ker((A - \lambda_i I_n)^r) \not\subset v^\perp$, but $Ker((A - \lambda_i I_n)^{r-1}) \subset v^\perp$. We saw in the proof of Theorem 3.4, that for any x in $Ker((A - \lambda_i I_n)^r) - Ker((A - \lambda_i I_n)^{r-1})$, such that $< v, x >> 0$, the program does not terminate. We fix such an x. Since both spaces $Ker((A - \lambda_i I_n)^r)$ and $Ker((A - \lambda_i I_n)^{r-1})$ are defined by linear equations with coefficients in L, there is a basis of $Ker((A - \lambda_i I_n)^r)$ with coefficients in L^n containing a basis of $Ker((A - \lambda_i I_n)^{r-1})$ with coefficients in L^n. It is easy to see that this implies that $L^n \cap [Ker((A - \lambda_i I_n)^r) - Ker((A - \lambda_i I_n)^{r-1})]$ is dense in $Ker((A - \lambda_i I_n)^r) - Ker((A - \lambda_i I_n)^{r-1})$, because L contains $\mathbb{Q}$ which is dense in $\mathbb{R}$. Hence, there is a sequence x_k in $L^n \cap [Ker((A - \lambda_i I_n)^r) - Ker((A - \lambda_i I_n)^{r-1})]$ which approaches x. In particular, $< v, x_k >> 0$ for k large enough. The program thus does not terminate on x_k when k is such that $\langle v, x_k \rangle > 0$. This shows that P_1 does not terminate on L^n. The fact that P_1 does not terminate on O_L is a trivial consequence of the fact that any element of L is the quotient of two elements of O_l. In particular, if P_1 does not terminate on $x \in L^n$, take $a > 0$ in O_L, such that $ax \in O_L^n$. Then the program does not terminate on ax. □ □

Let's see how Theorem 6.1 applies on our previous example.

Example 6.1. *For the program associated to the matrix $A = \begin{pmatrix} 0 & 1 \\ 1 & -2 \end{pmatrix}$ and the vector $v = (1, 0)^\top$, we get $L = \mathbb{Q}(\lambda^+) = \mathbb{Q}(\sqrt{2}) = \{a + b\sqrt{2}, a \in \mathbb{Q}, b \in \mathbb{Q}\}$. Its ring of integers is $O_L = \mathbb{Z}(\lambda^+) = \mathbb{Z}(\sqrt{2}) = \{a + b\sqrt{2}, a \in \mathbb{Z}, b \in \mathbb{Z}\}$. Theorem 6.1 asserts that, as the program $P(A, v)$ is non terminating, it is already non terminating on O_L^2. Indeed, Take x^+ as an initial value, then $x^+ = \begin{pmatrix} 1 \\ -1 + \sqrt{2} \end{pmatrix}$ belongs to O_L^2, and we saw that $P(A, v)$ does not terminate on x^+.* □

7 Discussion

The important papers [11, 12], treating homogeneous linear programs, can be seen at first as closely related to our results. The sufficient condition fully proved and established as our preliminary results in section 3.1, was first stated in [12]. On the other hand, the sufficient conditions proposed in [11, 12] are not necessary conditions for the termination of homogeneous linear programs and, thus, it is not obvious that one can obtain a direct encoding leading deterministically to a practicle algorithm. [12] can be divided in two parts. First, the interesting sketch of the proof of the sufficient condition leaves space for elaboration. We complete it in a solid mathematical way. We found obstacles (applying Brouwer's fixed point theorem to appropriate spaces, and having 0 in the closure of the orbit of a variable under the action of the transition matrix) not obvious. The second part provides a lengthy (3 reductions, a case analysis, long and costly symbolic computations) procedure inorder to check for termination. Also [11] is based on the approach proposed in [12], considering termination analysis over the integers. And similar arguments could be observed (e.g., the complex procedure appears lengthy and costly). In fact, it is not clear to us if those approaches give rise to simple and fast algorithms. Instead, we provide a clear statement which naturally provides a simple algorithm to check termination (showed by our examples), with much better complexity. Moreover, we show that it is enough to interpret the variable values over a countable number field (or its ring of integers) in order to determine program termination over the reals.

In our recent work about *asymptotically nonterminant values* (ANT) generation [13], we also provided new and efficient techniques to extend our results to general affine loop programs, *i.e.*, with several loop conditions. This discussion should be raised in another companion article where more practical details will be presented together with some experiments. Classical termination problems consider any possible initial values as done in the related literature. If our procedure returns terminant on any arbitrary initial values one can obviously have the same conclusion considering any initial precondition. The contributions presented in this paper are central to static data input analysis that we developed in those recent works. The generated ANT set can be used directly as preconfitions for termination or it can be intersected effciently with another given preconditions (provided by other static analysis methods for instance).

Our main results, Theorem 3.3 and its Corollary 3.1, with a direct encoding as Algorithm 1, together with and the results of Section 6, guaranteeing the symbolic computation while circumventing rounding errors, are evidences of the novelty of our approach.

8 Conclusions

We presented the *first necessary and sufficient condition* for the termination of linear homogeneous loop programs. This condition leads to a sound and complete procedure for checking termination for this class of programs. The analysis of our associated algorithms shows that our method operates in fewer computational steps than all known routines that support the mathematical foundations of previous methods. Section 6, and especially the example therein, introduces the important notion of the locus of initial variables values for which a linear program terminates. In this example, it allows us to decide if the program terminates on all rational initial variables values. Actually, these methods can be vastly generalized in order to tackle the termination problem of linear programs on rational initial values. However we suspect that it will involve some Galois theory as well as our results on asymptotically non terminant variable values, and so we leave this investigation for the near future.

References

[1] Cousot, P., Cousot, R.: Abstract interpretation and application to logic programs. Journal of Logic Programming **13**(2–3) (1992) 103–179

[2] Manna, Z.: Mathematical Theory of Computation. McGrw-Hill (1974)

[3] Cousot, P., Halbwachs, N.: Automatic discovery of linear restraints among variables of a program. In: Conference Record of the Fifth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Tucson, Arizona, ACM Press, New York, NY (1978) 84–97

[4] Sipma, H.B., Uribe, T.E., Manna, Z.: Deductive model checking. Form. Methods Syst. Des. **15**(1) (July 1999) 49–74

[5] Colón, M., Sipma, H.: Synthesis of linear ranking functions. In: Proceedings of the 7th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. TACAS 2001, London, UK, UK, Springer-Verlag (2001) 67–81

[6] Col'on, M.A., Sipma, H.B.: Practical methods for proving program termination. In: In CAV2002: Computer Aided Verification, volume 2404 of LNCS, Springer (2002) 442–454

[7] Bradley, A.R., Manna, Z., Sipma, H.B.: Linear ranking with reachability. In: In CAV, Springer (2005) 491–504

[8] Bradley, A.R., Manna, Z., Sipma, H.B.: Termination analysis of integer linear loops. In: In CONCUR, Springer-Verlag (2005) 488–502

[9] Dams, D., Gerth, R., Grumberg, O.: A heuristic for the automatic generation of ranking functions. In: Workshop on Advances in Verification. (2000) 1–8

[10] Podelski, A., Rybalchenko, A.: A complete method for the synthesis of linear ranking functions. In: VMCAI. (2004) 239–251

[11] Braverman, M.: Termination of integer linear programs. In: In Proc. CAV06, LNCS 4144, Springer (2006) 372–385

[12] Tiwari, A.: Termination of linear programs. In Alur, R., Peled, D., eds.: Computer Aided Verification, 16th International Conference, CAV 2004, Boston, MA, USA. Volume 3114 of Lecture Notes in Computer Science., Springer (2004) 70–82

[13] Rebiha, R., Matringe, N., Moura, A.V.: Automated generation of asymptotically non-terminant initial variable values for linear programs. Technical Report IC-14-03, Institute of Computing, University of Campinas (January 2014)

[14] Bradley, A.R., Manna, Z., Sipma, H.B.: Termination of polynomial programs. In: In VMCAI'2005: Verification, Model Checking, and Abstract Interpretation, volume 3385 of LNCS, Springer (2005) 113–129

[15] Chen, H.Y., Flur, S., Mukhopadhyay, S.: Termination proofs for linear simple loops. In: Proceedings of the 19th international conference on Static Analysis. SAS'12, Berlin, Heidelberg, Springer-Verlag (2012) 422–438

[16] Cousot, P.: Proving program invariance and termination by parametric abstraction, lagrangian relaxation and semidefinite programming. In: Sixth International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI'05), Paris, France, LNCS 3385, Springer, Berlin (January 17–19 2005) 1–24

[17] Cousot, P., Cousot, R.: An abstract interpretation framework for termination. SIGPLAN Not. **47**(1) (January 2012) 245–258

[18] Cook, B., Podelski, A., Rybalchenko, A.: Termination proofs for systems code. SIGPLAN Not. **41**(6) (June 2006) 415–426

[19] Ben-Amram, A.M., Genaim, S., Masud, A.N.: On the termination of integer loops. In: VMCAI. (2012) 72–87

[20] Ben-Amram, A.M., Genaim, S.: On the linear ranking problem for integer linear-constraint loops. In: Proceedings of the 40th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages. POPL '13, New York, NY, USA, ACM (2013) 51–62

[21] Rebiha, R., Matringe, N., Moura, A.V.: Necessary and sufficient condition for termination of linear programs. Technical Report IC-13-07, Institute of Computing, University of Campinas (February 2013)

[22] Rebiha, R., Matringe, N., Moura, A.V.: A complete approach for termination analysis of linear programs. Technical Report IC-13-08, Institute of Computing, University of Campinas (February 2013)

[23] Stein, W., Joyner, D.: SAGE: System for Algebra and Geometry Experimentation. ACM SIGSAM Bulletin, volume 39, number 2, pages 61–64 (2005)

Table 1: Experimental results on randomly generated linear loop programs

RandSet	#Loops	Fi	Dim	#T	#NT	CPU/s[T]	CPU/s[N]	CPU/s[total]
Set-1	500	$\mathbb{Z}$	3	152	348	10.02	8.79	18,24
Set-2	500	$\mathbb{Q}$	3	195	305	8.97	9.11	18.08
Set-3	500	$\mathbb{Z}$	3	233	267	15.07	12, 78	27,85
Set-4	500	$\mathbb{Q}$	3	223	277	12.49	10.42	22.91
Set-5	500	$\mathbb{Z}$	3	246	254	12.52	11.59	24,11
Set-6	500	$\mathbb{Q}$	3	222	278	13.30	10.35	23.66
Set-7	500	$\mathbb{R}$	4	122	378	27.8	16.51	44.31
Set-8	500	$\mathbb{Q}$	4	184	316	42,67	21.90	53.80
Set-9	500	$\mathbb{R}$	4	145	355	31.91	18.05	49.97
Set-10	500	$\mathbb{Q}$	4	171	329	41.16	22.37	63.54
Set-11	500	$\mathbb{Z}$	4	185	315	43.03	24.22	67.25
Set-12	500	$\mathbb{Q}$	4	176	324	40.36	19.95	60.32
Set-13	500	$\mathbb{R}$	5	183	317	126.24	66.95	193.20
Set-14	500	$\mathbb{Q}$	5	227	273	155.80	81.29	237.10
Set-15	500	$\mathbb{Z}$	5	178	322	103.90	43.47	146.57
Set-16	500	$\mathbb{Q}$	5	161	339	169.92	54.00	223.92
Set-17	500	$\mathbb{R}$	5	174	326	171.92	66.75	238.68
Set-18	500	$\mathbb{Q}$	5	158	342	174.91	70.32	254.24
Set-19	500	$\mathbb{Z}$	6	141	359	236.0	70.19	306.20
Set-20	500	$\mathbb{Q}$	6	173	327	387.80	105.69	493.50
Set-21	500	$\mathbb{Z}$	6	192	308	342.70	101.89	444,59
Set-22	500	$\mathbb{Q}$	6	188	312	352.40	165.41	517.81
Set-23	500	$\mathbb{Z}$	6	227	273	402.71	174.56	577.28
Set-24	500	$\mathbb{Q}$	6	184	316	385.00	190.94	575.94
Set-25	500	$\mathbb{Q}$	7	171	329	851.18	194.21	1044.39
Set-26	500	$\mathbb{Q}$	7	139	361	699.03	174.65	873.68
Set-27	500	$\mathbb{Q}$	7	166	334	876.62	238.94	1115.56