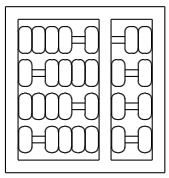

INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

On the Termination of Linear and Affine Programs over the Integers

Rachid Rebiha *Arnaldo V. Moura*
Nadir Matringe

Technical Report - IC-14-14 - Relatório Técnico

July - 2014 - Julho

The contents of this report are the sole responsibility of the authors.
O conteúdo do presente relatório é de única responsabilidade dos autores.

On the Termination of Linear and Affine Programs over the Integers

Rachid Rebiha* Arnaldo Vieira Moura† Nadir Matringe ‡

Abstract

The termination problem for affine programs over the *integers* was left open in [1]. For more that a decade, it has been considered and cited as a challenging open problem. To the best of our knowledge, we present here the most complete response to this issue: we show that termination for affine programs over $\mathbb{Z}$ is decidable under an assumption holding for almost all affine programs, except for an extremely small class of zero Lesbegue measure. We use the notion of *asymptotically non-terminating initial variable values* (ANT, for short) for linear loop programs over $\mathbb{Z}$. Those values are directly associated to initial variable values for which the corresponding program does not terminate. We reduce the termination problem of linear affine programs over the integers to the emptiness check of a specific ANT set of initial variable values. For this class of linear or affine programs, we prove that the corresponding ANT set is a semi-linear space and we provide a powerful computational methods allowing the automatic generation of these ANT sets. Moreover, we are able to address the conditional termination problem too. In other words, by taking ANT set complements, we obtain a precise under-approximation of the set of inputs for which the program does terminate.

1 Introduction

The *halting problem* is equivalent to the problem of deciding whether a given program will eventually terminate when running with a given input. The *termination problem* can be stated as follows: given an arbitrary program, decide whether the

*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.

program eventually halts for every possible input. Both problems are known to be undecidable, in general [2]. The *conditional termination problem* [3] asks for preconditions representing input data that will cause the program to terminate when run with such input data. In practice, this problem appears to be central for the verification of liveness properties that any well behaved and engineered system must guarantee. As it happens frequently, a program may terminate only for a specific set of input data values. Also, generating input data that demonstrates critical defects and vulnerabilities in programs allows for new looks at these liveness properties.

In the present work we address the termination and conditional termination problem over linear and affine while loop programs over the *integers*. In matrix terms, this class of programs can be expressed in the following form:

$$\texttt{while (F·x > b) \{x := A·x+c\}},$$

where A and F are matrices, b and c are vectors over the integers, and x is a vector of variables over $\mathbb{N}$ or $\mathbb{Z}$. The loop condition is a conjunction of linear or affine inequalities and the assignments to each of the variables in the loop instruction block are affine or linear forms. Static analysis and automated verification methods for programs presented in a more complex form can often be reduced to the study of a program expressed in this basic affine form [4].

Recent approaches for termination analysis of imperative loop programs have focused on partial decision procedures based on the discovery and synthesis of ranking functions [5, 6]. Several interesting techniques are based on the generation of *linear* ranking functions for linear loop programs [7, 8]. There are also effective heuristics [9, 6] and complete methods for the synthesis of *linear* ranking functions [10]. On the other hand, there are simple linear terminating loop programs for which it can be proved that there are no linear ranking functions.

On the problem of decidability results for termination of linear and affine programs, the work of Tiwari et al. [11] is often cited when treating linear programs over the *reals*. For linear programs over the *rationals* and *integers*, some of those theoretical results have been extended [1]. But the termination problem for *general affine* programs over the *integers* was left open in [1]. For more than a decade, it has been considered and cited as a challenging open problem [1, 11, 12, 13, 14]. This question was considered, but not completely answered, in [11, 1]. Recently, in [12], using strong results from analytic number theory and diophantine geometry, the authors were able to answer positively this question, but only when the corresponding transition matrices were restricted to a semi-simple form. In that work, ANT sets are not explicitly computed, because it is very hard by their approach — in fact, it is not proved they form semi-linear spaces, — and quantifier elimination is used. To the best of our knowledge, we present here the most complete response to this open problem: we show that termination for general affine programs over $\mathbb{Z}$ is decidable under an assumption holding for almost all affine programs except for a very small

class of zero Lesbegue measure. Our contribution is not limited to such theoretical decidability results, as we also provide efficient computational methods to decide termination and conditional termination for this class of programs. More specifically, in our approach the computation of the ANT set becomes simple, and it is described explicitly as a semi-linear space, without using quantifier elimination.

Concerning the termination and the conditional termination analysis problems, we could cite briefly the following recent developments. The framework presented in [15] is devoted to approaches establishing termination by abstract interpretation over the termination semantics. The approach exposed in [3] searches for non-terminating program executions. The recent literature on *conditional (non-)termination* narrows down to the works presented in [16, 3, 14]. The methods proposed in [16] allow for the generation of non-linear preconditions. In [3], the authors derived termination preconditions for simple programs — with only one loop condition — by guessing a ranking function and inferring a supporting assertion. Also, the interesting approach provided in [14] focuses mostly on proofs of decidability and consider several systems and models, but is restricted to two specific subclasses of linear relations. Despite tremendous progress over the years [1, 11, 17, 18, 15, 4, 19, 16, 3, 20], the problem of finding a practical, sound and complete method, *i.e.*, an encoding leading deterministically to an algorithm, for determining (conditional) termination remains very challenging. Some more closely related works, *e.g.* [16, 3, 14, 12, 1, 11], will be discussed in more details in Section 7.

Our initial investigations were reported in [21, 22] where we discussed termination analysis algorithms that ran in polynomial time complexity. Subsequent studies considered the set of *asymptotically non-terminating initial variable values* (ANT, for short) whose elements are directly related to input values for which the loop does not terminate [23]. In that work, we approached the problem of generating the ANT set for a restricted class of linear programs over the reals, with only one loop condition, and where the associated linear forms of the loop lead to diagonalizable systems with no complex eigenvalues. In [24], we showed how to compute the ANT set for linear or affine programs over $\mathbb{R}$. In that work we also successfully treated the case of linear or affine programs over $\mathbb{Z}$ in cases where the transition matrices admit a real spectrum. Here, we remove these restrictions. We show how to handle complex eigenvalues, linear affine programs over $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{N}$ or $\mathbb{Z}$, with conjunctions of several loop conditions, and where the system does not have to be diagonalizable. We thus drastically generalize the earlier results in [23, 24]. Further, we introduce new static analysis methods that compute ANT sets, and also yield a set of initial inputs values for which the program does terminate. This attests the innovation of our contributions, *i.e.*, none of the other mentioned works is capable of generating such critical information for non-terminating loops.

We summarize our contributions as follows, with all results rigorously stated and

proved:

On static input data analysis:

- We recall the important key concept of an ANT set [23] for linear loop programs over the integers. Theorems 3.1, 5.2 and 5.3 already show the importance of ANT sets. These results provide us with *necessary and sufficient conditions* for the termination of linear programs over the integers.
- For almost the whole class of linear, respectively affine, programs over $\mathbb{Z}$, namely those with transition matrix satisfying our Assumption 5.1, respectively Assumption 5.2, we prove that the set of asymptotically non-terminating inputs can be computed explicitly as a *semi-linear* space. Further, we show in Section 6 that almost all linear or affine programs belong to these classes, except for an extremely small specific class of zero Lesbegue measure. We are also capable of automatically generating a set of linear equalities and inequalities describing a semi-linear space that symbolically and exactly represents such ANT sets. See Theorem 4.1.
- Even if these results are mathematical in nature, they are easy to apply. In a practical static analysis scenario, one only needs to focus on ready-to-use generic formulas that represent the ANT sets for affine programs over $\mathbb{Z}$. See Definition 4.1, Eqs. (1), (2), (3) and (4). Such ANT set representations allow for practical computational manipulations — like union, intersection, complement and emptiness check, — and practical implementations.

On static termination and conditional termination analysis:

- We reduce the problem of termination for linear programs over the integers to the emptiness check of the corresponding ANT set. This characterization of terminating linear programs provides us with a deterministic computational procedure to *check program termination over* $\mathbb{Z}$, that is, we show that an affine program P is terminating if and only if P has an empty ANT set.
- Also, the ANT *complement* set is a precise under-approximation of the set of terminating inputs for the same program. This complement set gives rise to a loop precondition for termination. Thus, we obtain a computational methods for *conditional termination* analysis.

On decidability results for the termination problem over $\mathbb{Z}$:

- we obtain new decidability results for the program termination problem. Here, we successfully address the question left open in [1], namely, we settle the decidability problem for program termination in the case of affine

programs over the integers. Under our Assumption ($\mathcal{A}$) (see Fact 5.2), we prove that the termination problem for affine programs over $\mathbb{Z}$ is decidable.

- We provide a complete measure analysis of this assumption and show that our decidability results holds almost for all linear/affine programs overs $\mathbb{Z}$ (i.e., we prove that the class of affine programs not satisfying our assumption is of Lesbegue measure zero.).

Before concluding this section, we introduce a motivating example.

Example 1.1. (Motivating Example) *Consider the program:*

```
while(x-1/2y-2z>0){
 x:=-20x-9y+75z;
 y:=-7/20x+97/20y+21/4z;
 z:=35/97x+3/97y-40/97z;}
```

The initial values of x, y and z are represented, respectively, by the parameters u_1, u_2 and u_3. Our prototype outputs the following ANT set:

```
Locus of ANT:
    [[u1<-u2+3*u3]]OR[[u1==-u2+3*u3,-u3<u2]]OR[[u1==4*u3,u2==-u3,0<u3]].
```

The static input data analysis: *This semi-linear space represents symbolically all asymptotically initial values that are directly associated to initial values for which the program does not terminate.*

The conditional termination analysis: *The complement of this set is a precise under-approximation of the set of all initial values for which the program terminates.*

The termination analysis: *The problem of termination is reduced to the emptiness check of this ANT set.*

The paper is organized as follows. Section 2 presents basic results from linear algebra and also defines the computational model used to represent linear loop programs in matrix notation. Section 3 introduces the notion of *ANT* initial values, and presents the first important results for termination analysis. Section 4 provides an efficient computational method for generating a symbolic representation of the *ANT* set for linear homogeneous programs. This section also states the ready-to-use formulas representing symbolically and exactly the *ANT* sets for linear homogeneous programs. Section 5 reduces the study of generalized linear homogeneous and affine loop programs to that of linear homogeneous programs with a loop condition described by a single homogeneous inequality. Section 6 shows that our decidability result holds for all linear or affine programs, except for an extremely restricted class of zero measure programs. We provide a complete discussion in Section 7. Finally, Section 8 states our conclusions.

2 Preliminaries

In Subsection 2.1 we recall some classical concepts and results from linear algebra. In particular, we recall the Jacobian basis, in Theorem 2.1, and note a very useful basis, in Theorem 2.2. In subsection 2.2 we introduce the computational model for loop programs in matrix notation, and we provide an important classification for loop programs.

2.1 Linear Algebra

In the following, E will be a finite dimensional vector space over $\mathbb{R}$. Let A belong to $End_{\mathbb{R}}(E)$, the space of $\mathbb{R}$-linear maps from E to itself, and let E^* be the set of linear functionals in E, *i.e.*, of mappings from E to $\mathbb{R}$. In the sequel we will assume that f is a functional in E^*. We denote by $\mathcal{M}(p,q,\mathbb{R})$ the space of $p \times q$ matrices. When $p = q$ we may write $\mathcal{M}(p,\mathbb{R})$. If B is a basis of E, we denote by $Mat_B(A) \in \mathcal{M}(n,\mathbb{R})$ the matrix representation of A in the basis B. Let I_n be the identity matrix in $\mathcal{M}(n,\mathbb{R})$, and let $\mathbf{id_E}$ the identity of $End_{\mathbb{R}}(E)$.

We will denote by $Spec(A)$ the set of complex eigenvalues of A, by $Spec_{\mathbb{R}}(A)$ the set of real eigenvalues of A, and by $Spec_{>0}(A)$ the set of positive eigenvalues of A. In particular, we have

$$Spec_{>0}(A) \subset Spec_{\mathbb{R}}(A) \subset Spec(A).$$

We will also denote by $|Spec(A)|$, the set

$$\{|\mu|, \mu \in Spec(A)\},$$

and by $Spec_H(A)$, the intersection of $Spec(A)$ with the Poincaré upper half plane

$$H = \{z \in \mathbb{C},\ Im(z) > 0\}.$$

For $\lambda \in Spec_{\mathbb{R}}(A)$, we write E_λ for the characteristic subspace of A associated to λ, which is the kernel

$$Ker((A - \lambda I_d)^{d_\lambda}),$$

where d_λ is the multiplicity of λ in the characteristic polynomial χ_A of A. The non-real complex eigenvalues of A come into couples of conjugate complex numbers. If λ is such an eigenvalue, with $\overline{\lambda}$ its conjugate, we write $E_{\{\lambda,\overline{\lambda}\}}$ for

$$Ker[((A - \lambda I_d) \circ (A - \overline{\lambda} I_d))^{d_\lambda}],$$

where $\circ$ is the composition operator, and again d_λ is the multiplicity of λ in the characteristic polynomial of A. With these notations, we have the following direct sum decomposition:

$$E = \oplus_{\lambda \in Spec_{\mathbb{R}}(A)} E_\lambda \oplus_{\lambda \in Spec_H(A)} E_{\{\lambda,\overline{\lambda}\}}.$$

We recall the Jordan canonical basis theorem for A.

Theorem 2.1. *Let λ belong to $Spec_{\mathbb{R}}(A)$. There is a basis J_λ of E_λ such that $Mat_{J_\lambda}(A_{|E_\lambda}) = diag(U_{\lambda,1}, \ldots, U_{\lambda,r_\lambda})$ for a positive integer r_λ, where each $U_{\lambda,i}$ is of the form* $\begin{pmatrix} \lambda & 1 & & & & \\ & \lambda & 1 & & & \\ & & \ddots & \ddots & & \\ & & & \lambda & 1 & \\ & & & & \lambda & 1 \\ & & & & & \lambda \end{pmatrix}$.

For $\mu = a + ib = |\mu|e^{i\theta_\mu}$ a complex eigenvalue in $Spec_H(A)$, we denote by $s(\mu, \overline{\mu})$ the matrix

$$s(\mu, \overline{\mu}) = \begin{pmatrix} a & -b \\ b & a \end{pmatrix} = |\mu| r(\mu, \overline{\mu}),$$

where

$$r(\mu, \overline{\mu}) = \begin{pmatrix} cos(\theta_\mu) & -sin(\theta_\mu) \\ sin(\theta_\mu) & cos(\theta_\mu) \end{pmatrix}.$$

Similarly, we have the following theorem for u's restriction to $E_{\mu,\overline{\mu}}$.

Theorem 2.2. *Let μ belong to $Spec_H(A)$. There is a basis $J_{\mu,\overline{\mu}}$ of $E_{\mu,\overline{\mu}}$ such that*

$$Mat_{J_{\mu,\overline{\mu}}}(A_{|E_{\mu,\overline{\mu}}}) = diag(U_{\mu,\overline{\mu},1}, \ldots, U_{\mu,\overline{\mu},r_{\mu,\overline{\mu}}})$$

for a positive integer $r_{\mu,\overline{\mu}}$, where each $U_{\mu,\overline{\mu},i}$ is of the form

$$\begin{pmatrix} s(\mu,\overline{\mu}) & I_2 & & & & & \\ & s(\mu,\overline{\mu}) & I_2 & & & & \\ & & \ddots & \ddots & & & \\ & & & s(\mu,\overline{\mu}) & I_2 & & \\ & & & & s(\mu,\overline{\mu}) & I_2 & \\ & & & & & s(\mu,\overline{\mu}) & \end{pmatrix}.$$

2.2 Classification of Loop Programs

We recall, as it is standard in static program analysis, that a primed symbol x' refers to the next value of x after a transition is taken. First, we present *transition systems* as representations of imperative programs, and *automata* as their computational models.

Definition 2.1. *A* transition system *is given by $\langle x, L, \mathcal{T}, l_0, \Theta \rangle$, where $x = (x_1, ..., x_n)$ 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, for all $\tau \in \mathcal{T}$.*

A loop program is a transition system with a single location and a single transition, written simply as $\langle x, l, \langle l, l, q_\tau, \rho_\tau \rangle, l, \Theta \rangle$.

We will use the following matrix notation to represent loop programs and their transition systems. We also use simple and efficient procedures to captures the effects of sequential linear assignments into simultaneous updates.

Definition 2.2. *Let $P = \langle x, l, \langle l, l, q_\tau, \rho_\tau \rangle, l, \Theta \rangle$, with $x = (x_1, ..., x_n)$, be a loop program. We say that P is a* linear loop program *if:*

- *The transition guard is a conjunction 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 vector Fx is greater than the corresponding coordinate of vector b.*

- *The transition relation is a set of 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 loop program *$P(A, F, b, c)$ is defined as*

$$\texttt{while (F·x > b) \{x := A·x+c\}} \ .$$

We will use the following classification.

Definition 2.3. *From the more specific to the more general form:*

Homogeneous: *We denote by $P^{\mathbb{H}}$ the set of programs of the form*

$$P(A, f) : \texttt{while (f · x > b) \{x:=Ax\}} \ ,$$

where f is a $1 \times n$ row matrix corresponding to the loop condition, $b \in \mathbb{R}$, and $A \in \mathcal{M}(n, \mathbb{R})$ corresponds to the list of assignments in the loop.

Generalized Homogeneous: *We denote by $P^{\mathbb{G}}$ the set of programs of the form*

$$P(A, F) : \texttt{while (F x > 0) \{x:=Ax\}},$$

where F is a $(m \times n)$-matrix with rows corresponding to the m loop conditions. We will sometimes write $P(A, F) = P(A, f_1, \ldots, f_m)$, where the f_i's are the rows of F.

Affine: *We denote by $P^{\mathbb{A}}$ the set of programs of the form*

$$P(A, F, b, c) : \texttt{while (F x > b) } \{\texttt{x:=Ax+c}\} \ ,$$

for A and F as above, and $b, c \in \mathbb{R}^n$.

Example 2.1. *Consider the homogeneous program of Example 1.1. The sub-matrix* $A = \begin{pmatrix} -20 & -9 & 75 \\ 7 & 8 & -21 \\ -7 & -3 & 26 \end{pmatrix}$ *correspond to the simultaneous updates representing the sequential loop assignments and the vector $f = (1, -1/2, -2)^\top$ encodes the loop condition.*

In Section 5, we show that the termination analysis for the general class $P^{\mathbb{A}}$ can be reduced to termination for programs in $P^{\mathbb{H}}$.

3 The NT and ANT Sets

We present the new notion of *asymptotically non-terminating* (ANT) values of a loop program [23]. It will be central in the analysis of non-termination. We start with the definition of the ANT set and then give the first important result for homogeneous linear programs. We will extend these results in Section 5 to generalized linear homogeneous programs. Then, problem of termination analysis for the general class of linear programs will be reduced to the generation and the emptiness check of the ANT set for homogeneous linear programs.

Let E, $A \in End_{\mathbb{R}}(E)$ and $f \in E^*$ be as introduced in Section 2.1. In this section, we focus first on homogeneous programs

$$P(A, f) : \{\texttt{while (f·x>0) x:=A x}\}.$$

Given a basis B of E we write $\texttt{A} = Mat_B(A)$, $\texttt{f} = Mat_B(f)$, $\texttt{x} = Mat_B(x)$, and so on. From now on, we give definitions and statements in terms of programs involving linear maps, and let the reader infer the obvious adaptation for programs involving matrices. We start by giving the definition of the termination and non-termination for this class of programs.

Definition 3.1. *Let $P(A, f) \in P^{\mathbb{H}}$ and let $x \in E$ be an input for $P(A, f)$. We say that $P(A, f)$ terminates on x if and only if there exists some $k \geq 0$ such that $f(A^k(x)) \leq 0$; otherwise we say that $P(A, f)$ is non-terminating on x. If $K \subseteq E$, we say that $P(A, f)$ is terminating on K if and only if $P(A, f)$ terminates on every input $x \in K$. Further, program $P(A, f)$ is non-terminating (NT for short) if and only if it is non-terminating on some input $x \in E$.*

Thus, a program $P(A, f) \in P^{\mathbb{H}}$ is non-terminating if there is an input $x \in E$ such that $f(A^k(x)) > 0$ for all $k \geq 0$. We denote by $NT(P(A, f))$ the set of inputs $x \in E$ for which $P(A, f)$ is non-terminating.

Next, we introduce the important notion of an asymptotically non-terminating value [23].

Definition 3.2. *We say that $x \in E$ is an asymptotically non-terminating value for $P(A, f)$ if there exists some $k_x \geq 0$ such that $P(A, f)$ is non-terminating on $A^{k_x}(x)$. In this case, we will also say that $P(A, f)$ is ANT on x, or that x is ANT for $P(A, f)$. If $K \subseteq E$ we say that $P(A, f)$ is ANT on K if it is ANT on every $x \in K$. We will also say that $P(A, f)$ is ANT if it is ANT for some input $x \in E$.*

We denote by $ANT(P(A, f))$ the set of inputs $x \in E$ that are ANT for $P(A, f)$. The ANT set has a central role in the study and analysis of termination of program on any A-stable subset K of E, as we will show.

Example 3.1. *Consider again Example 1.1. We first note that the program terminates on $u = (-9, 3, -2)^\top$ because with this initial value no loop iteration will be performed as $fA^0u = -13/2$. It is also easy to check that $fA^1u = -5/2$, and that $fA^2u = 17.5$. In fact we have $fA^ku > 0$ for all $k \geq 2$, so that the program is non-terminating on $A^2u = (63, 3, 22)^\top$. We conclude that the initial value $u = (-9, 3, -2)^\top$ belongs to the ANT set.*

The following theorem already shows the importance of ANT sets: termination for linear programs is reduced to the emptiness check of the ANT set.

Theorem 3.1. *The program $P(\mathsf{A}, \mathbf{f})$ in $P^{\mathbb{H}}$ is NT if and only if it is ANT. More generally, if K is an A-stable subset of E, the program $P(A, f)$ is terminating on K if and only if $ANT(P(A, f)) \cap K$ is empty.*

Proof. It is clear that if $P(A, f)$ is NT, it is ANT as a NT value of $P(A, f)$ is, of course, also an ANT value. Conversely, if $P(A, f)$ is ANT, let x be an ANT value. Then $A^{k_x}(x)$ is a NT value of $P(A, f)$, and so $P(A, f)$ is NT. The assertion for A-stable subspaces of E is obvious, the proof being the same, as if $x \in K$ is ANT, we have $A^{k_x}(x) \in K$. □

The set of NT values is included in the ANT set, but the most important property of an ANT set resides in the fact that each of its elements gives an associated element in NT for the corresponding program. That is, each element x in the ANT set, even if it does not necessarily belong to the NT set, refers directly to initial values $A^{k_x}(x)$ for which the program does not terminate. Hence there is a number k_x of loop iterations, departing from the initial value x, such that $P(A, f)$ does not terminate on $A^{k_x}(x)$. This does not imply that x is NT for $P(A, f)$ because the program $P(A, f)$ could terminate on x by performing a number of loop iterations strictly smaller than k_x.

On the other hand, the ANT set is more than an over-approximation of the NT set, as it provide us with a deterministic and efficient way to decide termination.

Let ANT^c be the complement of the ANT set. It gives us an under approximation for the set of all initial values for which the program terminates.

Corollary 3.1. *Let $P(A, f)$ be in $P^{\mathbb{H}}$. Then $P(A, f)$ terminates on the complementary set $ANT^c(P(A, f))$.*

Proof. As $NT(P(A, f)) \subseteq ANT(P(A, f))$, passing to complementary sets gives the result. □

Theorem 3.1 provide a *necessary and sufficient conditions* for the termination of linear programs. Further, it allows for the reduction of the problem of termination for linear programs to the emptiness check of the corresponding ANT set. Also, Corollary 3.1 shows that ANT sets allow for the generation of initial variable values for which the loop program terminates. In the following section we prove that the ANT set is a semi-linear space, and we show how it can be exactly and symbolically computed.

4 Computation of ANT Sets for Homogeneous Programs

Let $P(A, f)$ be a program in $P^{\mathbb{H}}$. In this section we start with Assumption $(\mathcal{H})$ bellow, which will enable us to compute the sets $ANT(P(A, f))$ explicitly, and will also help us show that such sets are semi-linear subspaces of E. In Section 5 we show that a more general assumption reduces to this particular one, and in Section 6 we will show that this assumption is almost always satisfied, except for an extremely small class of programs.

Assumption 4.1 $(\mathcal{H})$**.** *In this section we will assume that $Spec_{\mathbb{R}}(A) = Spec_{>0}(A) \cup \{0\}$, and that if t is a positive eigenvalue of A then no other eigenvalue of A has the same module.*

We denote by $\mathbb{U}$ the set of complex numbers of module 1, *i.e.*, $\mathbb{U} = \{z \in \mathbb{C} \mid |z| = 1\}$. We will need the following lemma.

Lemma 4.1. *Let $u = (u_1, \ldots, u_r)$ be an element of $\mathbb{U}^r$, where $u_i \neq u_j$ and $u_i \neq \overline{u_j}$ when $i \neq j$, $1 \leq i, j \leq r$. Let s_k, $k \geq 0$, be as*

$$s_k = \sum_{i=1}^{n} (a_i u_i^k + \overline{a_i u_i}^k).$$

Then, either all a_i's are zero, or there is a $c > 0$ and there is an infinite number of k's such that $s_k < -c$.

Proof. According to Lemma 4 of [1], we know that either s_k is constantly zero, or we are in the second situation of the statement. But if s_k is constantly zero, then by Dedekind's theorem on linear independence of characters applied to $\mathbb{Z}$, all the a_i's are zero. □

If τ is a positive real number in $|Spec(A)| - Spec_{>0}(A)$, we set

$$A_\tau = \{\mu \in Spec_H(A), |\mu| = \tau\},$$

and

$$\Sigma_\tau = \oplus_{\mu \in A_\tau} E_{\mu,\overline{\mu}}.$$

We also set

$$U_\tau = \{u_\mu = \mu/\tau, \mu \in A_\tau\}.$$

The following proposition is as a consequence of the Jordan basis Theorems 2.1 and 2.2, in Section 2.1.

Proposition 4.1. *If τ is a positive real number in $|Spec(A)| - Spec_{>0}(A)$, then for x_τ in Σ_τ, the quantity $f(A^k(x_\tau))$ is of the form*

$$[\sum_{j=0}^{d_\tau - 1}(\sum_{u_\mu \in U_\tau} a_{\mu,j}(x_\tau)u_\mu^k + \overline{a_{\mu,j}(x_\tau)u_\mu^k})k^j]\tau^k,$$

where d_τ is the maximum of the integers $dim_{\mathbb{R}}(E_{\mu,\overline{\mu}})/2$, for $\mu \in A_\tau$, and the $a_{\mu,j}$'s are $\mathbb{R}$-linear maps from Σ_τ to $\mathbb{C}$, which can be computed explicitly. If t is a positive eigenvalue of A, for x_t in E_t, the quantity $f(A^k(x_t))$ is of the form

$$(\sum_{i=0}^{d_t - 1} \alpha_{t,i}(x_t)k^i)t^k,$$

where d_t is the dimension of E_t, and the $\alpha_{t,i}$'s are $\mathbb{R}$-linear maps from E_t to $\mathbb{R}$, which can be computed explicitly.

We are now going to describe the $ANT(P(A, f))$ sets as semi-linear spaces of E. We note that the linear maps $\alpha_{t,i}$ and $a_{\mu,j}$, in Proposition 4.1, can be computed easily. In our previous work [24], we showed how these linear maps are computed efficiently for programs over the reals and for programs over $\mathbb{Z}$, when the induced matrix A had a real spectrum. Here, the computation of $\alpha_{t,i}$ and $a_{\mu,j}$ remains similar to those described in [24], Sections 7 and 8.

We first introduce the following subsets of E.

Definition 4.1. *For $t \in Spec_{>0}(A)$, and l between 0 and $d_t - 1$, we define the sets $S_{t,l}$ to be the sets of elements x in E which satisfy:*

- *For $\tau > t$ in $|Spec(A)|$,*
  - *if $\tau \notin Spec(A)$, then for all $\mu \in A_\tau$ and $j \in \{0, \ldots, d_\tau - 1\}$:*

$$a_{\mu,j}(x_\mu) = 0. \tag{1}$$

  - *if $\tau \in Spec(A)$, then for all $i \in \{0, \ldots, d_\tau - 1\}$*

$$\alpha_{\tau,i}(x_\tau) = 0. \tag{2}$$

- *For all i between $l+1$ and $d_t - 1$,*

$$\alpha_{t,i}(x_t) = 0. \tag{3}$$

- *Finally we have the inequalities:*

$$\alpha_{t,l}(x_t) > 0. \tag{4}$$

We can now state the main result of this section, describing the generic formulas representing exactly and symbolically the ANT sets.

Theorem 4.1. *The set $ANT(P(A, f))$ is the disjoint union of the sets $S_{t,l}$, for $t \in Spec_{>0}(A)$, and $l \in \{0, \ldots, d_t - 1\}$. In other words, considering the set $\Delta_S = \{(t, l) \mid t \in Spec_{>0}(A), l \in \{0, \ldots, d_t - 1\}\}$ we have*

$$ANT(P(A, f)) = \bigvee_{(t,l) \in \Delta_S} S_{t,l}.$$

Proof. First, if x belongs to $S_{t,l}$ then, by assumption, the sequence $fA^k x$ will be asymptotically equivalent to

$$t^k \alpha_{t,l}(x_t) k^l,$$

which grows without bound. Hence, x belongs to $ANT(P(A, f))$.

Conversely, suppose that x belongs to none of the $S_{t,l}$ sets. Let τ be the highest absolute value among the eigenvalues of A, such that for $\tau' > \tau$, τ' being the module of an eigenvalue of A, we have

$$a_{\mu,j}(x_\mu) = 0$$

for all $\mu \in A_{\tau'}$ and $j \in \{0, \ldots, d_{\tau'} - 1\}$ when $\tau' \notin Spec(A)$, and

$$\alpha_{\tau',i}(x_{\tau'}) = 0$$

for all $i \in \{0, \ldots, d_{\tau'} - 1\}$ when $\tau' \in Spec(A)$. Then,

- If $\tau = 0$, we get $f(A^k(x)) = f(A^k(x_0))$, which is constantly zero for k large enough. Hence, x is not in $ANT(P(A, f))$.

- If $\tau > 0$, we have two possibilities, depending on whether τ is in $Spec(A)$, or not.

  – When $\tau \notin Spec(A)$, let l be the highest integer between 0 and $d_\tau - 1$ such that $a_{\tau,l}(x_\tau)$ is nonzero. We know, from Lemma 4.1, that for an infinite number of k's, the sum

  $$s_k = \sum_{u_\mu \in U_\tau} a_{\mu,l}(x_\tau)u_\mu^k + \overline{a_{\mu,l}(x_\tau)u_\mu^k}$$

  is smaller than a negative number $-c$, which is independent of k. As the integers k grow it follows, from Proposition 4.1, that fA^kx will be equivalent to fA^kx_τ. By the choice of l, the latter itself be equivalent to

  $$s_k k^l \tau^k \leq -ck^l\tau^k,$$

  which decreases without bound. Hence, fA^kx will be negative for an infinite number of k's, and thus x does not belong to $ANT(P(A, f))$.

  – Nw assume $\tau = t \in Spec_{>0}(A)$, and let l be the highest integer between 0 and $d_t - 1$ such that $\alpha_{t,l} \neq 0$. Then, as x is not in $S_{t,l}$, we must have

  $$\alpha_{t,l}(x_t) < 0.$$

  But fA^kx is equivalent to

  $$\alpha_{t,l}(x_t)t^k$$

  when k grows according to Proposition 4.1. Hence, fA^kx decreases without bound, and so x is not in $ANT(P(A, f))$. This completes the proof.

□

In the next section we generalize these results to programs in the classes $P^{\mathbb{G}}$ and P^{A}. We show that the problem of generating ANT sets for linear and affine programs reduces to the computation ANT sets for specific homogeneous programs under Assumption $(\mathcal{H})$.

5 Termination over $\mathbb{Z}$ for Linear and Affine Programs

In this section, we extend our methods to linear and affine programs. For each of these program classes, the ANT set generation problem is reduced to the computation of ANT sets of corresponding homogeneous programs under Assumption $(\mathcal{H})$.

For $f_1, \ldots, f_r$ a family of elements in E^*, $b \in \mathbb{R}^r$, and c a vector of E, we consider the *affine program* $P(A, F, b, c) = P(A, (f_i)_{i=1,\ldots,r}, b, c) \in P^{\mathsf{A}}$:

$$P(A, (f_1, \ldots, f_r), b, c) : \texttt{while (} \bigwedge_{1 \leq i \leq r} \texttt{fi(x)>bi) \{x:=Ax+c\}}.$$

We will also consider the *linear program* $P(A, F) = P(A, (f_i)_{i=1,\ldots,r})$, where

$$P(A, F) = P(A, (f_1, \ldots, f_r)) = P(A, (f_1, \ldots, f_r), 0, 0).$$

5.1 ANT sets for generalized homogeneous programs

First , we remove some restrictions on A. We denote by $R(A)$ the set of nonzero eigenvalues of A with arguments a rational multiple of 2π, *i.e.*,

$$R(A) = \{\lambda \in Spec(A),\ Arg(\lambda) \in 2\pi\mathbb{Q}\} = \{\lambda \in Spec(A),\ \exists n \in \mathbb{N},\ \lambda^n > 0\}.$$

Assumption 5.1 $(\mathcal{G})$**.** *For any eigenvalue λ of A in $R(A)$, if $\mu \in Spec(A) - \{0\}$ is such that $|\mu| = |\lambda|$, then μ is equal to λ up to a root of unity in $\mathbb{C}$, i.e. if λ and μ in $Spec(A) - \{0\}$ are such that $|\lambda| = |\mu|$, then either both are in $R(A)$, or none is. In other words, two nonzero eigenvalues with the same module both have an argument which is either a rational multiple of 2π, or none has.*

From now on, we suppose that A satisfies Assumption $(\mathcal{G})$. As the rational numbers are a negligible set of $\mathbb{R}$, we see that for a generic matrix A, the set $R(A)$ is empty. Hence, almost all matrices A in $\mathcal{M}(n, \mathbb{R})$ satisfy Assumption $(\mathcal{G})$. In Section 6, we will confirm this fact, and actually show more precisely that the set of matrices satisfying Assumption $(\mathcal{G})$ contains a dense open set of total measure, *i.e.*, whose complement is of measure zero.

First, we show that we can reduce the computation of the ANT set for an homogeneous program $P(A, f) \in P^{\mathbb{H}}$, when A satisfies Assumption $(\mathcal{G})$, to the intersection of the ANT sets of programs $P(G, g) \in P^{\mathbb{H}}$, with G satisfying Assumption $(\mathcal{H})$. This reduction technique is also used in [12]. First, we notice that an appropriate power of A satisfies Assumption $(\mathcal{H})$.

Proposition 5.1. *Let Q be the set defined by*

$$Q = \{\mu/|\mu|,\ \mu \in R(A)\}.$$

If N is the lcm *of the orders of elements of Q, then A^N satisfies Assumption $(\mathcal{H})$.*

Proof. If r belongs to $Spec_{>0}(A^N)$, let λ be an eigenvalue of A^N, such that $|\lambda| = r$. As λ is in $Spec(A^N)$, it is equal to μ^N for some $\mu \in Spec(A)$, which is in fact in $R(A)$ as $\mu^N > 0$. Similarly, $r = \mu'^N$ for $\mu' \in Spec(A)$. But $\mu'^N = r > 0$, and so $\mu' \in R(A)$. As

$$|\mu| = |\mu'| = r^{1/N},$$

by Assumption $(\mathcal{G})$, μ' is also in $R(A)$. But then, by the definition of N, we have $\mu'^N > 0$, and so $\lambda = \mu'^N = r$. Thus, the second part of Assumption $(\mathcal{H})$ is satisfied.

Moreover, if A^N had a negative eigenvalue λ, again it would be of the form $\lambda = \mu^N$. But then we would have $\mu^{2N} = \lambda^2 > 0$ and so, by the definition of N, we would get $(\mu/|\mu|)^N = 1$. That is, $\lambda = |\mu|^N$, which is absurd. Hence, A^N also satisfies the first part of Assumption $(\mathcal{H})$. □

We recall that, in the previous section, we showed that if G satisfies assumption $(\mathcal{H})$, then for any $g \in E^*$ the set $ANT(P(G, g))$ is semi-linear, and we computed it explicitly. Now, we show how to compute $ANT(P(A, f))$.

Theorem 5.1. *We have* $ANT(P(A, f)) = \cap_{l=0}^{N-1} ANT(P(A^N, fA^l))$.

Proof. It is clear that

$$ANT(P(A, f)) \subset \cap_{l=0}^{N-1} ANT(P(A^N, fA^l)).$$

Conversely, if x belongs to $\cap_{l=0}^{N-1} ANT(P(A^N, fA^l))$. Then for every l between 0 and $N-1$, there is $m_{x,l}$, such that $k \geq m_{x,l}$, which gives $fA^{kN+l}x > 0$. Taking

$$m_x = max_{l \in \{0,\ldots,N-1\}} m_{x,l},$$

we have

$$k \geq m_x \Rightarrow f(A^k(x)) > 0,$$

that is, $x \in ANT(P(A, f))$. This proves the equality. □

Proposition 5.1 guarantees that the matrix A^N satisfies Assumption $(\mathcal{H})$. Considering a program $P(A, f) \in P^{\mathbb{H}}$, with A satisfying Assumption $(\mathcal{G})$, Theorem 5.1 shows that the $ANT(P(A, f))$ set is the intersection of the $ANT(P(A^N, fA^l))$ sets, with A^N satisfying Assumption $(\mathcal{H})$. This handles the case of linear homogeneous programs with one loop condition under assumption $(\mathcal{G})$. Now, as in [24], we reduce the computation of the ANT set of a generalized homogeneous program to that of a homogeneous program.

Consider $P(A, F) = P(A, (f_i)_{i=1,\ldots,r})$ in $P^{\mathbb{G}}$. We start with the following lemma on non-terminating values.

Definition 5.1. *The value x is NT for $P(A, F)$ in $P^{\mathbb{G}}$ if and only if it is NT for all $P(A, f_i)$, with $i \in \{1, \ldots, r\}$.*

Now, we define ANT values for such programs.

Definition 5.2. *We say that x is ANT for $P(A,F)$ if there exists k_x such that for all $i \in \{1,\dots,r\}$ we have $f_i(A^k(x)) > 0$ for $k > k_x$, that is, if x is ANT for all programs $P(A,f_i)$.*

Again we have the following easily proved but important lemma.

Lemma 5.1. *A program $P(A,F)$ is NT if and only if it is ANT, that is, $ANT(P(A,F)) \neq \emptyset$.*

Proof. If x belongs to $NT(P(A,F))$, then it belongs to $ANT(P(A,F))$. Conversely, if x belongs to $ANT(P(A,F))$, then for some k, by definition, $A^k(x)$ belongs to $NT(P(A,F))$. In particular, both sets are empty or non empty together, which proves the claim. □

We can now express the ANT set of programs in $P^{\mathbb{G}}$ as the intersection of ANT sets from corresponding programs in $P^{\mathbb{H}}$.

Proposition 5.2. *Let $f_1,\dots,f_r$ be linear forms on E, then one has*

$$ANT(P(A,(f_1,\dots,f_r))) = \cap_{i=1}^{r} ANT(P(A,f_i)).$$

Proof. If x is in $ANT(P(A,(f_1,\dots,f_r)))$, there is $k \geq 0$ such that $f_i A^l(x) > 0$ for $l \geq k$, for all i, hence x belongs to every set $ANT(P(A,f_i))$. Conversely, if x belongs to $\cap_{i=1}^{r} ANT(P(A,f_i))$, then for each i, there is $k_i \geq 0$, such that $l \geq k_i$ implies $f_i A^l(x) > 0$. Take $k = max_i(k_i)$, then $l \geq k$ imples that for every i, $f_i A^l(x) > 0$, i.e. x belongs to $ANT(P(A,(f_1,\dots,f_r)))$. □

5.2 ANT sets for affine programs over $\mathbb{Z}$

First, we define the notion of ANT values for affine programs.

Definition 5.3. *Let $P(A,F,b,c)$ be an affine program in $P^{\mathbb{A}}$. For $x = x_0 \in \mathbb{R}^n$, denote by x_1 the vector $Ax + c$ and, recursively, let $x_k = Ax_{k-1} + c$, $k \geq 1$. We say that a vector x is ANT for $P(A,F,b,c)$ if there is some k_x such that $k \geq k_x$ implies $Fx_k > b$. We denote by $ANT(P(A,F,b,c))$ the set of ANT input values of $P(A,F,b,c)$.*

Consider the affine program $P(A,F,b,c) = P(A,(f_1,\dots,f_r),b,c)$. We denote by E' the vector space $E \oplus \mathbb{R}$. We denote by A' the linear map from E' to itself defined by

$$A' : x + t \mapsto (Ax + tc) + t,$$

and let f'_i, $1 \leq i \leq r$, be the linear form on E' defined by

$$f'_i : x + t \mapsto f_i(x) - tb_i,$$

and let $f'_{r+1} : x + t \mapsto t$. Finally, for x in E, we set $x' = x + 1$ in E'. As we have

$$Spec(A') = Spec(A) \cup \{1\},$$

we notice at once the following fact.

Fact 5.1. *A' satisfies Assumption ($\mathcal{G}$), if and only if A satisfies it, and no eigenvalue in $Spec(A) - R(A)$ has module 1.*

We make this conclusion explicit.

Assumption 5.2. *($\mathcal{A}$) Let $P(A, F, b, c)$ be an affine program in $P^{\mathbb{A}}$. We say that A satisfies Assumption ($\mathcal{A}$) when it satisfies Assumption ($\mathcal{G}$) and no eigenvalue in $Spec(A) - R(A)$ has module 1.*

When working with an affine program $P(A, F, b, c)$, also written as $P(A, (f_1, \ldots, f_r), b, c)$, we will assume that A satisfies Assumption ($\mathcal{A}$).

Proposition 5.3. *The input x is in the set $ANT(P(A, (f_1, \ldots, f_r), b, c))$ if and only if input x' is in the set $ANT(P(A', (f'_1, \ldots, f'_r)))$.*

Proof. Fix B a basis of E, and let $\mathbf{A} = Mat_B(A) \in \mathcal{M}(n, \mathbb{R})$, $\mathbf{F} \in \mathcal{M}(r, n, \mathbb{R})$ the matrix with rows equal to the $Mat_B(f_i)$'s, $\mathbf{b} = Mat_B(b)$ in $\mathcal{M}(1, r, \mathbb{R})$, and let $\mathbf{c} = Mat_B(c)$. Let B' be the basis of $E \oplus \mathbb{R}$, with first vectors $(e_i, 0_{\mathbb{R}})$, for e_i in B, and last vector $(0_E, 1)$. Now let $\mathbf{A'} = Mat_{B'}(A') \in \mathcal{M}(n+1, \mathbb{R})$ and $\mathbf{F'} \in \mathcal{M}(r+1, n+1, \mathbb{R})$ the matrix with rows $Mat_{B'}(f'_i)$. Clearly, we have $\mathbf{A'} = \begin{bmatrix} \mathbf{A} & \mathbf{c} \\ 0 & 1 \end{bmatrix}$, and $\mathbf{F'} = \begin{bmatrix} \mathbf{F} & -\mathbf{b} \\ 0 & 1 \end{bmatrix}$. To say that $(x, 1)^\top$ is ANT for $P(\mathbf{A'}, \mathbf{F'})$ means that there exists k_x, such that when $k \geq k_x$, we get $\mathbf{F'}\mathbf{A'}^k \cdot (x, 1)^\top > 0$. We define x_k by induction, as $x_0 = x$, and $x_{k+1} = Ax_k + c$. But as $\mathbf{A'} \cdot (x, 1)^\top = \begin{pmatrix} \mathbf{A}x + \mathbf{c} \\ 1 \end{pmatrix} = (x_1, 1)^\top$, by induction, we obtain $\mathbf{A'}^k \cdot (x, 1)^\top = (x_k, 1)^\top$, $k \geq 1$. But $\mathbf{F'}\mathbf{A'}^k \cdot (x, 1)^\top = \mathbf{F'} \cdot (x_k, 1)^\top = \begin{pmatrix} \mathbf{F}x_k - \mathbf{b} \\ 1 \end{pmatrix}$. Hence, $\mathbf{F'}\mathbf{A'}^k \cdot (x, 1)^\top > 0$ is equivalent to $\mathbf{F}x_k > \mathbf{b}$, and the result follows. □

Proposition 5.3 shows that the generation of the ANT set for a program in $P^{\mathbb{A}}$ reduces to the generation of the ANT set for an associated program in $P^{\mathbb{G}}$, and that reduces to the computation of ANT sets for corresponding homogeneous programs in $P^{\mathbb{H}}$. These two reduction provide us with computational methods for the automatic generation of ANT sets for affine programs under Assumption ($\mathcal{A}$). Now, we can state the following termination result for generalized homogeneous and affine programs over $\mathbb{Z}$.

Theorem 5.2. *Let $A \in \mathcal{M}(n, \mathbb{Z})$ be a matrix over the integers associated to loop instructions.*

- *Then $P(A, (f_1, \ldots, f_r))$ terminates on $\mathbb{Z}^n$ if and only if*

$$ANT(P(A, f_1, \ldots, f_r)) \cap \mathbb{Z}^n = \emptyset,$$

- *$P(A, (f_1, \ldots, f_r), b, c)$ terminates on $\mathbb{Z}^n$ if and only if*

$$ANT(P(A', (f'_1, \ldots, f'_r))) \cap \mathbb{Z}^n \times 1 = \emptyset.$$

Proof. We know that the computation of the $ANT(P(A, (f_1, \ldots, f_r)))$ sets reduces to the intersection of the $ANT(P(A, f_i))$ sets, with $1 \leq i \leq r$, by Proposition 5.2. For the $ANT(P(A, f_i))$ sets we apply Theorem 3.1, with $K = \mathbb{Z}^n$, and thus establish the first assertion. We saw that $x \in K \subseteq E$ is ANT (resp. NT) for $P(A, F, b, c)$ if and only if $x' = (x, 1)^\top$ is ANT (resp. NT) for $P(A', F')$, with $A' = \begin{bmatrix} A & c \\ & 1 \end{bmatrix}$, and $F' = \begin{bmatrix} F & -b \\ 0 & 1 \end{bmatrix}$. We apply Theorem 3.1 with $K' = \{x', x \in \mathbb{Z}^n\}$, which is A'-stable. □

The following corollary states the main decidability result for the termination problem for affine programs over the integers.

Corollary 5.1. *Under Assumption $(\mathcal{G})$, the termination over $\mathbb{Z}^n$ of programs in the form $P(A, f_1, \ldots, f_r)$ is decidable. Under Assumption $(\mathcal{A})$, the termination of programs $P(A, f_1, \ldots, f_r, b, c)$ over $\mathbb{Z}^n$ is decidable.*

Proof. We appeal to a result in [25], which asserts that it can be decided if a convex semi-algebraic subspace of $\mathbb{R}^n$, contains an element of the lattice $\mathbb{Z}^n$. We apply it to the subspace $ANT(P(A, f_1, \ldots, f_r))$ of $\mathbb{R}^n$ in the first case. In the second case, we apply it to the image of the subspace

$$\mathbb{R}^n \times \{1\} \cap ANT(P(A', f'_1, \ldots, f'_r))$$

of $\mathbb{R}^n \times \{1\}$ under the canonical projection from $\mathbb{R}^n \times \{1\}$ to $\mathbb{R}^n$. □

Corollary 5.1 provides the most complete response to the termination problem left open in [1].

6 Matrices Satisfying Assumptions $(\mathcal{G})$ or $(\mathcal{A})$

In this section, we show that Assumptions $(\mathcal{G})$ or $(\mathcal{A})$ are almost always satisfied.

Theorem 6.1. *The set of matrices A in $\mathcal{M}(n,\mathbb{R})$ satisfying Assumption $(\mathcal{G})$ contains a dense open subset of $\mathcal{M}(n,\mathbb{R})$, and of total Lebesgue measure in $\mathcal{M}(n,\mathbb{R})$. The same assertion is true for matrices satisfying Assumption $(\mathcal{A})$.*

Proof. We consider the set U of $\mathcal{M}(n,\mathbb{R})$, of semi-simple — that is, diagonalizable over $\mathbb{C}$, — matrices with distinct eigenvalues. It is the complement set of zeros of

$$P : A \mapsto disc(\chi_A, \chi'_A),$$

where *disc* stands for the discriminant. Thus, *disc* is dense, open, and of total measure. We denote by W the open subset of $\mathbb{C}^n$, consisting of n-uples $(z_1,\dots,z_n)$, which satisfy $z_i \neq z_j$ when $i \neq j$.

Let $\sigma_i(z_1,\dots,z_n)$ denote the coefficient of X^i in $(X-z_1)\dots(X-z_n)$. It is well known that the map σ from W to $\mathbb{C}^n[X]$, defined by

$$\sigma : (z_1,\dots,z_n) \mapsto (\sigma_0(z_1,\dots,z_n),\dots,\sigma_{n-1}(z_1,\dots,z_n)),$$

is a local diffemorphism, as its Jacobian at z equals, up to the sign, the product

$$\prod_{i<j}(z_i - z_j).$$

We are going to show the set of matrices in U, which do not have two non-conjugate eigenvalues with the same absolute value is open, dense, and of total measure in $\mathbb{R}^n$. As this set is contained in the set of matrices satisfying $(\mathcal{G})$, this will prove our first assertion.

Let $\mathbb{C}_{n,1}[X]^{reg}$ denote the set of monic polynomials of $\mathbb{C}_n[X]$ with distinct roots, and let P be such a polynomial. Number its roots as $(z_1(P),\dots,z_n(P)) \in W$. Then, there is an open neighborhood $\mathcal{N}_\mathcal{P}$ of P in $\mathbb{C}_{n,1}[X]^{reg}$ such that, for Q in $\mathcal{N}_\mathcal{P}$, one can number the roots of Q as $(z_1(Q),\dots,z_n(Q)) \in W$, and

$$R : Q \mapsto (z_1(Q),\dots,z_n(Q))$$

is a smooth diffeomorphism from $\mathcal{N}_\mathcal{P}$ to its open image $R(\mathcal{N}_\mathcal{P}) \subset W$. Hence, if P belongs to $\mathbb{R}_{n,1}[X]^{reg} = \mathbb{C}_{n,1}[X]^{reg} \cap \mathbb{R}_n[X]$, we have that

$$\mathcal{N}_\mathcal{P}{}^r = \mathcal{N}_\mathcal{P} \cap \mathbb{R}_n[X] \subset \mathbb{R}_{n,1}[X]^{reg}$$

is a submanifold of $\mathcal{N}_\mathcal{P}$, $R(\mathcal{N}_\mathcal{P}{}^r)$ is a submanifold of $R(\mathcal{N}_\mathcal{P})$, and the restriction of R to $\mathcal{N}_\mathcal{P}{}^r$ is thus a smooth diffeomorphism to its image $R(\mathcal{N}_\mathcal{P}{}^r)$. In fact, it is easy to see what $R(\mathcal{N}_\mathcal{P}{}^r)$ looks like.

We denote by $B(u, \epsilon)$ the open ball of radius $\epsilon > 0$ around the complex number u. Suppose that $(z_1(P), \ldots, z_n(P))$ is ordered in such a way that $z_1(P), \ldots, z_a(P)$ are real, and the other roots come in b couples of conjugate complex numbers $(z_i(P), z_{i+1}(P))$ with $z_{i+1}(P) = \overline{z_i(P)}$ and $n = a + 2b$. Then, one can choose $\mathcal{N}_\mathcal{P}$ such that for some positive ϵ , $R({\mathcal{N}_\mathcal{P}}^r)$ is diffeomorphic to the product

$$]z_1(P) - \epsilon, z_1(P) + \epsilon[\ \times \ \ldots \ \times \ ]z_a(P) - \epsilon, z_a(P) + \epsilon[\\ \times B(z_{a+1}, \epsilon) \times B(z_{a+3}, \epsilon) \cdots \times B(z_{a+2b-1}, \epsilon).$$

In particular, the intersection of $R({\mathcal{N}_\mathcal{P}}^r)$ with the set $|z_i| = |z_j|$ when i and j are such that $z_i(P)$ and $z_j(P)$ are not conjugate, is a hypersurface of $R({\mathcal{N}_\mathcal{P}}^r)$.

Finally, as the map

$$A \in U \mapsto \chi_A \in \mathbb{R}_{n,1}[X]^{reg}$$

is submersive everywhere then, the set of matrices in U, which have two distinct non conjugate eigenvalues with the same module, is locally the union of at most $n(n-1)/2$ hypersurfaces. In particular, its complementary set is open, dense, and of total measure in U, hence in $\mathbb{R}^n$. We have proved our first assertion.

The second assertion's proof is completely similar. □

7 Discussion

In this section we note some related works. Then we summarize some of our previous results along similar lines, and list the main contributions presented here.

7.0.1 Related work:

Concerning the *termination analysis* for affine programs over the reals, rationals and the integers, we reduced the problem to the emptiness check of the generated ANT sets. By so doing, we obtained a characterization of terminating linear programs which allows for a practical and computational procedure. In [1, 11], the authors focused on the decidability of the termination problem for linear loop programs. Also, the techniques in [1] are based on the approach in [11], but now considering termination analysis over the rationals and integers for *homogeneous programs* only. But the termination problem for *general affine* programs over the integers is left open in [1].

Recently, in [12], considering the ANT set and a technique similar to our approach previously proposed in [23, 24], the authors were able to answer this question for programs with semi-simple matrices, using strong results from analytic number theory, and diophantine geometry. By contrast, in [12] the author focus on decidability results, and the ANT set is not explicitly computed there. In fact, the ANT set is referred to as a semi-algebraic set and the use of quantifier elimination techniques

is suggested. In this work, although we also considered the termination problem, we addressed a more general problem, namely, the *conditional termination* problem of generating static sets of terminating and non-terminating inputs. We provide efficient computational methods allowing for the exact computation and symbolic representation of the ANT sets for affine loop programs over $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{Z}$, and $\mathbb{N}$. The ANT sets generated by our approach can be seen as a precise over-approximation for the set of non-terminating inputs. We use "precise" in the sense that $NT \subseteq ANT$ and all elements in ANT, even those not in NT, are directly associated with non-terminating values, modulo a finite numbers of loop iterations. The, possibly infinite, complement of an ANT set is also a "precise" under-approximation of the set of terminating inputs, as it provides terminating input data entering the loop at least once.

Our method differs from those proposed in [3], as we do not use the synthesis of ranking functions.

The methods proposed in [16] can provide non-linear preconditions, but we always generate semi-linear sets as precondition for termination, which facilitates the static analysis of liveness properties.

The approach in [14] considers first octagonal relations and the associated class of formulae representing weakest recursive sets. It also suggests the use of quantifier elimination techniques and algorithms, which would require an exponential running time complexity of order $O(n^3 \cdot 5^n)$, where n is the number of variables. They also consider the conditional termination problem for restricted subclasses of linear affine relations, where the associated matrix has to be diagonalizable and with all non-zero eigenvalues of multiplicity one. They also identify other classes where the generated precondition would be non-linear.

The experiments in [26], involving handwritten programs, are handled successfully by our algorithm presented in a companion article [24], more oriented towards static program analysis. The strength and the practical efficiency of the approach is shown by our experiments dealing with a large number of larger linear loops. In [24], we present several details related to the application of the theoretical contributions exposed here. Our prototype was tested and the average time to generate the ANT over 9000 randomly generated loops was 0.75 seconds. In this experiment, the associated matrices were triangularizable, with a number of variables between 3 and 15, and a number of conjunctions forming the loop condition between 1 to 4. In this more static program analysis applied work, we used examples from [26, 3, 14, 1, 20, 10].

7.0.2 Our prior work:

We list here the points most relevant to the present discussion.

- In [21, 22] we provided new termination analysis algorithms that ran in polynomial time complexity.

- We considered the set of *asymptotically non-terminating initial variable values* for the first time in [23]. In that work we generated the ANT set for a restricted class of linear programs over the reals, with only one loop condition, and where the associated linear forms of the loop lead to diagonalizable systems.

- In [24] we showed how to automatically generate the ANT sets for linear and affine programs. In that work, we also handled the case of linear or affine programs over $\mathbb{Z}$ with transition matrices admitting a real spectrum. It is the first substantial contribution on termination of linear program over the integers. Here, we removed these restrictions. In [24], we also treated the case of matrices with a real spectrum. But if that is the case, if two distinct eigenvalues have the same module, one is the opposite of the other, that is, they are equal up to the root of unity. In particular, in this case, Assumption $(\mathcal{G})$ is always satisfied, and so the results obtained here fully generalize those obtained in [24].

7.0.3 The main contributions:

The central contributions presented in this article are listed below.

- Our criteria for termination over stable subspaces allowed us to show that termination for linear or affine programs over $\mathbb{Z}$ is decidable for almost the whole class of such programs.

- We proved that the ANT set is a semi-linear space, and we provided a computational method allowing for their automatic generation.

- We rigorously proved that our assumption holds for almost all linear or affine programs by showing that the excluded programs forms an extremely small set of zero Lebesgue measure.

- Our main results, Theorems 3.1, 4.1, 5.1, 5.2, 5.3, 5.2, 5.1, and 6.1, are evidences of the novelty of our approach.

8 Conclusions

In terms of decidability results, we provide the most complete response to the termination problem for linear or affine programs over the integers. We reduced the termination problem of linear, affine programs over $\mathbb{Z}$ to the emptiness check of the ANT set of corresponding homogeneous linear programs. Then, we proved that these sets are semi-linear spaces which are easy to compute and manipulate.

These theoretical contributions are mathematical in nature with proofs that are quite technical. We showed, however, that these results can be directly applied in

practical ways. One can rely the ready-to-use formulas representing the ANT set provided in this article.

Also, any static program analysis technique could incorporate, by a simple and direct instantiation, the generic ready-to-use formulas representing the preconditions for termination and non-termination.

References

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

[2] Turing, A.M.: On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society **2**(42) (1936) 230–265

[3] Cook, B., Gulwani, S., Lev-Ami, T., Rybalchenko, A., Sagiv, M.: Proving conditional termination. In: Proceedings of the 20th International Conference on Computer Aided Verification. CAV '08, Berlin, Heidelberg, Springer-Verlag (2008) 328–340

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

[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, Springer-Verlag (2001) 67–81

[6] Colón, 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] 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

[12] Ouakine, J., Pinto, J.S., Worrell, J.: On termination of integer linear loops. Technical report, http://arxiv.org/abs/1407.1891. (July 2014)

[13] Ben-Amram, A.M., Genaim, S., Masud, A.N.: On the termination of integer loops. ACM Trans. Program. Lang. Syst. **34**(4) (December 2012) 16:1–16:24

[14] Bozga, M., Iosif, R., Konecený, F.: Deciding conditional termination. In: Proceedings of the 18th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. TACAS'12, Berlin, Heidelberg, Springer-Verlag (2012) 252–266

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

[16] Gulwani, S., Srivastava, S., Venkatesan, R.: Program analysis as constraint solving. In: Proceedings of the 2008 ACM SIGPLAN Conference on Programming Language Design and Implementation. PLDI '08, New York, NY, USA, ACM (2008) 281–292

[17] 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

[18] 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

[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] Rebiha, R., Matringe, N., Moura, A.V.: Generating asymptotically non-terminant initial variable values for linear diagonalizable programs. In Kovacs, L., Kutsia, T., eds.: SCSS 2013. Volume 15 of EPiC Series., EasyChair (2013) 81–92

[24] Rebiha, R., Matringe, N., Moura, A.V.: Generating asymptotically non-terminant initial variable values for linear programs. Technical Report IC-14-09, Institute of Computing, University of Campinas (June 2014)

[25] Khachiyan, L., Porkolab, L.: Computing integral points in convex semi-algebraic sets. In: In FOCS'1997. 162–171

[26] Ganty, P., Genaim, S.: Proving termination starting from the end. In: CAV. (2013) 397–412