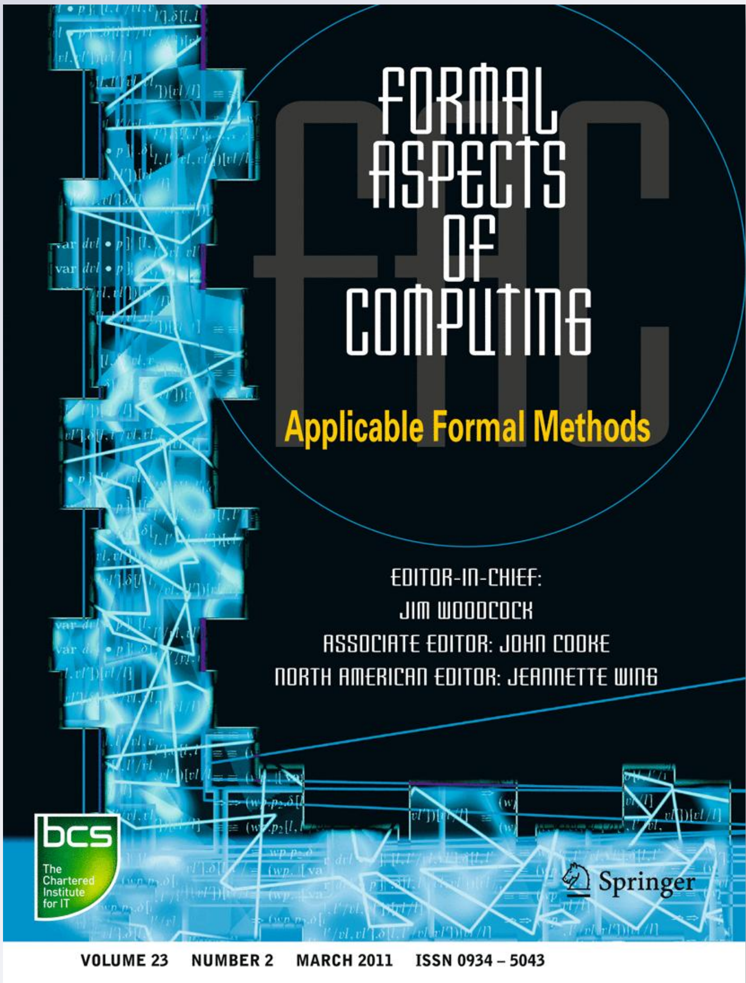
Symbolic decision procedure for termination of linear programs

Formal Aspects of Computing
Applicable Formal Methods

ISSN 0934-5043
Volume 23
Number 2

Form Asp Comp (2009)
23:171-190
DOI 10.1007/
s00165-009-0144-5



Springer

DOI 10.1007/s00165-009-0144-5
BCS © 2009
Formal Aspects of Computing (2011) 23: 171–190

Formal Aspects of Computing

Symbolic decision procedure for termination of linear programs

Bican Xia[1], Lu Yang[2], Naijun Zhan[3] and Zhihai Zhang[1]

[1] LMAM, School of Mathematical Sciences, Peking University, Beijing, China. E-mails: xbc@math.pku.edu.cn; infzzh@math.pku.edu.cn
[2] Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China. E-mail: lyang@sei.ecnu.edu.cn
[3] Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China. E-mail: znj@ios.ac.cn

Abstract. Tiwari proved that the termination of a class of linear programs is decidable in Tiwari (Proceedings of CAV'04. Lecture notes in computer science, vol 3114, pp 70–82, 2004). The decision procedure proposed therein depends on the computation of *Jordan forms*. Thus, people may draw a wrong conclusion from this procedure, if they simply apply floating-point computation to compute Jordan forms. In this paper, we first use an example to explain this problem, and then present a symbolic implementation of the decision procedure. Thus, the rounding error problem is therefore avoided. Moreover, we also show that the symbolic decision procedure is as efficient as the numerical one given in Tiwari (Proceedings of CAV'04. Lecture notes in computer science, vol 3114, pp 70–82, 2004). The complexity of former is $\max\{O(n^6), O(n^{m+3})\}$, while that of the latter is $O(n^{m+3})$, where n is the number of variables of the program and m is the number of its Boolean conditions. In addition, for the case when the characteristic polynomial of the assignment matrix is irreducible, we design a more efficient symbolic algorithm whose complexity is $\max(O(n^6), O(mn^3))$.

Keywords: Linear programs, Termination, Symbolic computation, Numerical computation

1. Introduction

Floating-point computation is a source of run-time errors of embedded software. The stories of the Ariane 5 launcher [Ari96] and the Patriot missile [Ske92] are evidence for this. Therefore verification of embedded software has to take into account the rounding error issue of floating-point computation. The well-known static program analyzer, ASTRÉE, meets this challenge, includes rounding error analysis in its abstract interpretations [Min05], and achieves success to some extent. However there are many interesting verification algorithms and techniques which have not paid to this issue sufficient attention yet. For example, Tiwari in [Tiw04] proved that the termination of the following loops on the reals is decidable.

$$\mathtt{P}_1: \quad \textbf{while} \quad (B\mathbf{x} > \mathbf{b}) \quad \{\mathbf{x} := A\mathbf{x} + \mathbf{c}\},$$

where A is an $n \times n$ matrix, B is an $m \times n$ matrix, and $\mathbf{x}$, $\mathbf{b}$ and $\mathbf{c}$ are vectors. $B\mathbf{x} > \mathbf{b}$ is a conjunction of strict linear inequalities which is the loop condition, while $\mathbf{x} := A\mathbf{x} + \mathbf{c}$ is interpreted as updating the values of $\mathbf{x}$ by $A\mathbf{x} + \mathbf{c}$ simultaneously and not in any sequential order. We say $\mathtt{P}_1$ terminates if it terminates on all initial values.

Correspondence and offprint requests to: B. Xia, E-mail: xbc@math.pku.edu.cn

The termination problem of $\mathtt{P}_1$ is reduced to that of the following homogeneous loop in [Tiw04]

$$\mathtt{P}_2: \qquad \textbf{while} \quad (B\mathbf{x} > 0) \quad \{\mathbf{x} := A\mathbf{x}\}.$$

A key step of the decision procedure in [Tiw04] is to compute the *Jordan form* of the matrix A so that one can have a diagonal description of A^n. In [Tiw04] it was proved that if the Jordan form of A is $A^* = Q^{-1}AQ$ and $B^* = BQ$, then $\mathtt{P}_2$ terminates if and only if

$$\mathtt{P}_2^*: \qquad \textbf{while}\ (B^*\mathbf{x} > 0)\ \{\mathbf{x} := A^*\mathbf{x}\}$$

terminates. This idea is natural, but, if we use floating-point computation routines to calculate the Jordan form in a conventional way, the errors of floating-point computation may lead to a wrong conclusion. To see this point, let us consider the following example.

Example 1 Let

$$A = \begin{bmatrix} 2 & -3 \\ -1 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & b \\ -1 & b \end{bmatrix},$$

where $b = -\dfrac{1127637245}{651041667} = -\sqrt{3} + \epsilon \approx -1.732050807$, with $\epsilon = \sqrt{3} - \dfrac{1127637245}{651041667} > 0$. Determine whether

$$\mathtt{Q}_1: \qquad \textbf{while}\ (B\mathbf{x} > 0)\ \{\mathbf{x} := A\mathbf{x}\}$$

is terminating.

According to the conventional method, in order to compute the Jordan form of A we have to calculate the eigenvalues of A by using floating-point computation, say, through the package `linalg` (or `LinearAlgebra`) in Maple 11. The approximate eigenvalues of A are 3.732050808 and 0.267949192 (both take 10 decimal digits of precision). Hence, the Jordan form of A is

$$A^* = Q^{-1}AQ = \begin{bmatrix} 3.732050808 & 0 \\ 0 & 0.267949192 \end{bmatrix}$$

where

$$Q = \begin{bmatrix} 0.5 & 0.5 \\ -0.2886751347 & 0.2886751347 \end{bmatrix}.$$

Use the same package of Maple 11 to calculate

$$B^* = BQ = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & -1.0 \end{bmatrix}.$$

Then, the loop $\mathtt{Q}_1$ is terminating if and only if the following loop $\mathtt{Q}_2$ terminates,

$$\mathtt{Q}_2: \qquad \textbf{while}\ (B^*\mathbf{x} > 0)\ \{\mathbf{x} := A^*\mathbf{x}\}.$$

Obviously, $(A^*)^n = \begin{bmatrix} 3.732050808^n & 0 \\ 0 & 0.267949192^n \end{bmatrix}$. If we let $\mathbf{x} = [1, -1]^{\mathrm{T}}$, after n times of iteration,

$$(A^*)^n\mathbf{x} = [3.732050808^n, -0.267949192^n],$$

where $\mathbf{v}^{\mathrm{T}}$ stands for the transpose of the vector $\mathbf{v}$. And the loop condition is

$$B^*(A^*)^n\mathbf{x} = [3.732050808^n, 0.267949192^n] > [0, 0],$$

which is always true for all n. Therefore, $\mathtt{Q}_1$ is not terminating.

However, this conclusion is not correct. Let us see how the floating-point computation leads us to the wrong result. The Jordan form of A is indeed (by symbolic computation)

$$J = P^{-1}AP = \begin{bmatrix} 2+\sqrt{3} & 0 \\ 0 & 2-\sqrt{3} \end{bmatrix},$$

where

$$P = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{6}\sqrt{3} & \frac{1}{6}\sqrt{3} \end{bmatrix}$$

and, in order to obtain B^*, we should compute BP instead of BQ symbolically as

$$BP = \begin{bmatrix} \frac{1}{2} - \frac{b}{6}\sqrt{3} & \frac{1}{2} + \frac{b}{6}\sqrt{3} \\ -\frac{1}{2} - \frac{b}{6}\sqrt{3} & -\frac{1}{2} + \frac{b}{6}\sqrt{3} \end{bmatrix} = \begin{bmatrix} 1 - \frac{\epsilon}{6}\sqrt{3} & \frac{\epsilon}{6}\sqrt{3} \\ -\frac{\epsilon}{6}\sqrt{3} & -1 + \frac{\epsilon}{6}\sqrt{3} \end{bmatrix}$$
$$= \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}.$$

Therefore $m_{12} > 0$, $m_{21} < 0$. However, when we use floating-point computation, these two entries (m_{12} and m_{21}) are evaluated to 0 (in Maple 11 with Digits 10). That is why we obtain wrong result by floating-point computation.

One may guess that if we evaluate BP rather than BQ through floating-point computation routines, we may obtain a more precise approximation. Unfortunately, it is not true. In fact using floating-point computation to compute BP we will still get some strange results. For example, computing BP by Maple 11 with Digits 10 outputs the following matrix

$$\begin{bmatrix} 1.0 & -1.0 \cdot 10^{-10} \\ 1.0 \cdot 10^{-10} & -1.0 \end{bmatrix}.$$

It is totally wrong, because, comparing with the signs of m_{12} and m_{21} in the above, m_{12} is negative and m_{21} positive.

To handle the above problem, in this paper, we develop a symbolic decision procedure for the termination of linear programs $\mathtt{P}_1$. The general framework of our procedure is quite similar to that of [Tiw04], but we re-implement the two key steps, i.e., computing Jordan normal form of A and generating linear constraints. In Tiwari's decision procedure, the two steps are implemented numerically, in contrast that in this paper we will give different algorithms based on symbolic computation to implement the two steps. Thus, our decision procedure can avoid errors caused by floating-point computation. According to [Tiw04], it is easy to reduce the termination problem of $\mathtt{P}_1$ to that of $\mathtt{P}_2$, therefore we only concentrate on $\mathtt{P}_2$ instead of $\mathtt{P}_1$ in the rest. The basic idea of our approach is: Firstly, classify and then represent eigenvalues of A symbolically, afterwards symbolically compute a set of eigenvectors and generalized eigenvectors of A, which can form an invertible matrix P such that $P^{-1}AP$ is the Jordan normal form of A.[1] Thus, the termination problem of $\mathtt{P}_2$ is symbolically reduced to that of $\mathtt{P}_2^*$. Secondly, we present a symbolic decision procedure to determine whether $\exists \mathbf{x}\, \forall n \in \mathbb{N}. B^*(A^*)^n \mathbf{x} > 0$, i.e., whether P_2^* terminates. Furthermore, by complexity analysis, we show that our symbolic decision procedure is as efficient as the one given in [Tiw04]. The complexity of the algorithm is $\max\{O(n^6), O(n^{m+3})\}$, where n is the number of variables of $\mathtt{P}_2$ and m is the number of the Boolean conditions of $\mathtt{P}_2$. In contrast, the complexity of the decision procedure developed in [Tiw04] is $O(n^{m+3})$.

In addition, we also consider the case when the characteristic polynomial of A is irreducible. A much simpler and more efficient decision algorithm is invented by solving a univariate semi-algebraic system. The complexity of the algorithm for this case is $\max(O(n^6), O(mn^3))$.

For saving space, we assume the reader is familiar with linear algebra; otherwise, please refer to [HoK71].

In order to improve the readability, we adopt the following *convention* on the use of variables: In what follows, we use $\mathcal{J}, \mathcal{B}, \ldots$ possibly with subscripts to stand for sets of matrices; $A, B, \ldots$ possibly with subscripts for matrices; $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{u}, \mathbf{v}, \ldots$ possibly with subscripts for vectors; and $x, y, z, a, b, c, \ldots$ possibly with subscripts for reals or integers.

The paper is organized as follows: Sect. 2 reviews basic theories of semi-algebraic systems and the tool DISCOVERER, which are used in the paper; Sect. 3 is devoted to designing a symbolic decision algorithm for $\mathtt{P}_2$ which is demonstrated by an example; We prove the correctness of the algorithm in Sect. 4; Sect. 5 analyzes

[1] Precisely, we do not calculate a complete invertible transition matrix of A, just its sub-matrix related to the positive eigenvalues of A.

the complexity of the symbolic decision procedure, and concludes that the symbolic decision procedure is as efficient as the numerical one given in [Tiw04]; In Sect. 6, we establish a necessary and sufficient condition for the termination of programs $\mathtt{P}_2$ when the characteristic polynomial of A is irreducible, and then design a symbolic decision algorithm with the complexity $\max(O(n^6), O(mn^3))$; Finally we conclude the paper and discuss related issues in Sect. 7.

2. Semi-algebraic systems and DISCOVERER

In this section, we will briefly review semi-algebraic systems and the tool DISCOVERER, which will be used in the later.

Definition 1 A semi-algebraic system (SAS) is a system of

$$\begin{cases} p_1(\mathbf{u},\mathbf{x}) = 0, \ldots, p_r(\mathbf{u},\mathbf{x}) = 0, \\ g_1(\mathbf{u},\mathbf{x}) \geq 0, \ldots, g_k(\mathbf{u},\mathbf{x}) \geq 0, \\ g_{k+1}(\mathbf{u},\mathbf{x}) > 0, \ldots, g_t(\mathbf{u},\mathbf{x}) > 0, \\ h_1(\mathbf{u},\mathbf{x}) \neq 0, \ldots, h_m(\mathbf{u},\mathbf{x}) \neq 0, \end{cases} \tag{1}$$

where $\mathbf{u} = (u_1, \ldots, u_d), \mathbf{x} = (x_1, \ldots, x_s)$, $r, s \geq 1$, $d, m \geq 0$, $t \geq k \geq 0$ and all p_i's, g_i's and h_i's are polynomials over $\mathbb{Q}$, the set of rationals. An SAS of the form (1) is called *parametric* if $d \neq 0$, otherwise *constant*.

For a constant SAS S, interesting questions are how to compute the number of real solutions of S, and if the number is finite, how to compute these real solutions. For a parametric SAS, the interesting problem is so-called *real solution classification*, that is to determine the condition on the parameters such that the system has the prescribed number of distinct real solutions, possibly infinite.

Yang et al. developed theories on how to classify real roots of parametric SASs in [YHZ96, Yan99, YaX05] and isolate real roots of constant SASs in [XiY02]. The core of the theories is the generalized *Complete Discrimination System* (CDS) in [YHZ96]. A computer algebra tool named DISCOVERER [Xia07] has been developed using Maple to implement these theories. Comparing with other well-known computer algebra tools for solving problems in real algebra like REDLOG [DoS97] and QEPCAD [CoH91], DISCOVERER has distinct features in the above two aspects.

The main features of DISCOVERER include

Real Solution Classification of Parametric Semi-algebraic Systems

For a parametric SAS S of the form (1) and an argument N, where N is one of the following three forms:

- a non-negative integer b;
- a range $b..c$, where b, c are non-negative integers and $b < c$;
- a range $b.. + \infty$, where b is a non-negative integer,

DISCOVERER provides **tofind** and **Tofind** to determine the conditions on $\mathbf{u}$ such that the number of the distinct real solutions of S equals to N if N is an integer, otherwise falls in the scope N.

Real Solution Isolation of Constant Semi-algebraic Systems

For a constant SAS S of the form (1), if S has only a finite number of real solutions, DISCOVERER can determine the number of distinct real solutions of S, say n, and moreover, can find out n disjoint cubes with rational vertices in each of which there is only one solution. In addition, the width of the cubes can be less than any given positive real. The two functions are realized by calling **nearsolve** and **realzeros**, respectively.

Since Maple 13, DISCOVERER has been integrated into the RegularChains [CLL08] package of Maple. Please see the corresponding help pages of Maple for how to call the functions of DISCOVERER in Maple.

3. A symbolic decision procedure

In this section, we present a symbolic decision procedure for the termination of $\mathtt{P}_2$. The procedure consists of two steps: Firstly, we symbolically determine the real eigenvalues of matrix A and represent them symbolically too.

Then compute a set of eigenvectors and generalized eigenvectors in terms of its eigenvalues. These eigenvectors and generalized eigenvectors form a matrix P such that $A^* = P^{-1}AP$ is the Jordan normal form of A. Thus, the termination problem of $\mathtt{P}_2$ is reduced to that of $\mathtt{P}_2^*$; Secondly, we design a symbolic procedure to determine if $\mathtt{P}_2^*$ terminates. According to Tiwari's result, the termination behavior of $\mathtt{P}_2^*$ is determined in the state space corresponding to the positive eigenvalues of A. So, in the first step, we only need to consider the problems related to the positive eigenvalues of A.

3.1. Computing (generalized) eigenvectors symbolically

It is well-known that each column of the transition matrix of A (i.e., P in the above) is either an eigenvector or a generalized eigenvector of A, see for instance pp. 82–86 of [MiM82]. In the following, we develop an algorithm for computing (generalized) eigenvectors of A in terms of its symbolic eigenvalues.

Let $D(\lambda) = f_1(\lambda)^{i_1} \cdots f_k(\lambda)^{i_k}$ be the characteristic polynomial of A, where each f_j is an irreducible polynomial[2] in $\mathbb{Q}[\lambda]$, $i_j > 0$ and $d_j = \text{degree}(f_j)$ ($\deg(f_j)$ for short) for $j = 1, \ldots, k$, and $d_1 i_1 + \cdots + d_k i_k = n$. Suppose λ is a real root of f_1 (i.e., a real eigenvalue of A) and $\mathbf{v} = (v_1, \ldots, v_n)^{\mathrm{T}}$ is an eigenvector corresponding to λ, then

$$(A - \lambda I)\mathbf{v} = 0. \tag{2}$$

Without loss of generality, we assume $d_1 > 1$, i.e., λ is an algebraic number of degree greater than 1.[3] By applying the so-called *fraction-free Gaussian elimination* (FFGE) to $A - \lambda I$ with modulus $f_1(\lambda) = 0$, (2) can be solved symbolically. We will demonstrate the procedure via the following example.

Example 2 Let $A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 3 & 0 & 0 & -3 & 0 \\ 0 & 1 & 0 & 3 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$. Thus, the characteristic polynomial of A is $D(\lambda) = (\lambda - 1)(\lambda^2 - 3)^2$.

Denote the unique irrational positive root of $D(\lambda)$ by λ_0. Applying FFGE to the following matrix (mod $\lambda_0^2 - 3 = 0$)

$$\begin{bmatrix} -\lambda_0 & 1 & 1 & 0 & 0 \\ 3 & -\lambda_0 & 0 & -3 & 0 \\ 0 & 1 & -\lambda_0 & 3 & 0 \\ 0 & 0 & 1 & -\lambda_0 & 0 \\ 0 & 0 & 0 & 0 & 1-\lambda_0 \end{bmatrix}$$

an eigenvector related to λ_0 in terms of λ_0 is computed as follows: Firstly, we replace the second row by the first row multiplied by 3 plus the second row multiplied by λ_0 and obtain

$$\begin{bmatrix} -\lambda_0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 3 & -3\lambda_0 & 0 \\ 0 & 1 & -\lambda_0 & 3 & 0 \\ 0 & 0 & 1 & -\lambda_0 & 0 \\ 0 & 0 & 0 & 0 & 1-\lambda_0 \end{bmatrix}.$$

Then, we exchange the second and the third rows, subsequently replace the fourth row by the fourth row plus the third row multiplied by $-1/3$ and obtain

$$\begin{bmatrix} -\lambda_0 & 1 & 1 & 0 & 0 \\ 0 & 1 & -\lambda_0 & 3 & 0 \\ 0 & 0 & 3 & -3\lambda_0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-\lambda_0 \end{bmatrix}.$$

---

[2] A polynomial is said to be *irreducible* over a field, say rational numbers, if it cannot be factorized into non-trivial polynomials over the field. Otherwise *reducible*. In general, we can check whether a rational polynomial is reducible over rational numbers by computer algebra tools such as **Maple** and **Mathematica**.

[3] If $d_1 = 1$, λ is a rational number and thus the (generalized) eigenvectors corresponding to λ can be solved easily.

One can easily obtain the solutions of $\mathbf{v}$ to $(A - \lambda_0 I)\mathbf{v} = 0$ in terms of λ_0 as $v_1 = v_4, v_2 = 0, v_3 = \lambda_0 v_4, v_4 = v_4, v_5 = 0$. If let $v_4 = 1$, we get an eigenvector $\mathbf{v} = [1, 0, \lambda_0, 1, 0]^{\mathrm{T}}$.

From the above example, it is easy to see that, for a given A and an eigenvalue λ such that $f_j(\lambda) = 0$, we can use the FFGE algorithm to compute a set of linear independent eigenvectors that form a base of the eigenspace related to λ. We denote the algorithm by $\mathtt{EV}(A, f_j, \lambda)$.

A vector $\mathbf{v}$ is called a *generalized eigenvector* of rank j of A related to λ if

$$(A - \lambda I)^j \mathbf{v} = 0,\ (A - \lambda I)^{j-1}\mathbf{v} \neq 0,\ \ j \geq 1. \tag{3}$$

Computing a generalized eigenvector is obviously a similar problem as solving (2) and can also be solved by the FFGE algorithm. For a given A and an eigenvalue λ, the algorithm for computing a generalized eigenvector of rank j related to λ is denoted by $\mathtt{GEV}(A, f_i, \lambda, j)$, where $f_i(\lambda) = 0$.

For a square matrix A with all entries in $\mathbb{R}$, in [MiM82] an algorithm is presented on how to numerically compute A's eigenvalues, their relevant eigenvectors and generalized eigenvectors, and A's invertible transition matrix P such that $P^{-1}AP$ is its Jordan normal form. Similarly, by exploiting FFGE we can extend the algorithm to symbolically compute A's eigenvalues, their relevant eigenvectors and generalized eigenvectors, and A's invertible transition matrix P such that $P^{-1}AP$ is its Jordan normal form. We will denote the resulting algorithm by JordanBlocks in this paper. Thus, using JordanBlocks, one can compute the set of Jordan blocks $[J_1, \ldots, J_k]$ related to λ, denoted by $\mathcal{J}_\lambda$. Accordingly, one can find a related block set $[B_1, B_2, \ldots, B_k]$ which is a submatrix of B that corresponds to the block set $\mathcal{J}_\lambda$, denoted by $\mathcal{B}_\lambda$, called *condition block set* related to λ.

3.2. Main algorithm

In this subsection, we shall give a symbolic algorithm to determine whether $\mathtt{P}_2$ terminates symbolically.

First of all, we outline the decision procedure for $\mathtt{P}_2$ given in [Tiw04] in order to not only ease the reader to understand the difference between the two decision procedures, but also facilitate us to present our symbolic algorithm as the algorithm adopts the skeleton of Tiwari's procedure.

The first step is to reduce the termination problem of $\mathtt{P}_2$ to that of $\mathtt{P}_2^*$.

Suppose the Jordan blocks of A are $J_1, \ldots, J_l$ and therefore BP is divided into $B^* = [B_1, \ldots, B_l]$ accordingly. Thus, $\mathtt{P}_2^*$ can be reformulated as

$$\mathtt{P}_3: \qquad \textbf{while}\ (B_1\mathbf{y}_1 + B_2\mathbf{y}_2 + \cdots + B_l\mathbf{y}_l > 0)\{\mathbf{y}_1 := J_1\mathbf{y}_1;\ \ldots;\ \mathbf{y}_l := J_l\mathbf{y}_l\}.$$

Let $S^+ = \{i \mid 1 \leq i \leq l \wedge J_i$ is a Jordan block corresponding to a positive eigenvalue$\}$.

The second step of the procedure given in [Tiw04] is to prove that $\mathtt{P}_3$ terminates iff the program

$$\mathtt{P}_4: \qquad \textbf{while}\ \left(\sum_{i \in S^+} B_i\mathbf{y}_i > 0\right)\{\mathbf{y}_i := J_i\mathbf{y}_i;\ \ \text{for } i \in S^+\}$$

terminates.

For brevity, we assume all the eigenvalues of A are positive in the above. Therefore, the kth loop condition of $\mathtt{P}_4$ is $\mathbf{b}_{1k}\mathbf{y}_1 + \mathbf{b}_{2k}\mathbf{y}_2 + \cdots + \mathbf{b}_{lk}\mathbf{y}_l > 0$, where $\mathbf{b}_{ik}$ stands for the kth row of B_i. Given an input $\mathbf{y}(0)$ with $\mathbf{y}(0)^T = (\mathbf{y}_1(0)^T, \ldots, \mathbf{y}_l(0)^T)$, the requirement that the kth condition still holds after the nth iteration for $\mathbf{y}(0)$ can be expressed as

$$\mathbf{b}_{1k}\mathbf{y}_1(n) + \mathbf{b}_{2k}\mathbf{y}_2(n) + \cdots + \mathbf{b}_{lk}\mathbf{y}_l(n) > 0,$$

where $\mathbf{y}_j(n) = J_j^n\mathbf{y}_j(0)$. After expanding and collecting the above formula, let $C_{kij}\mathbf{y}(0)$, a linear expression of $\mathbf{y}(0)$, denote the coefficient of the term $\binom{n}{j-1}\lambda_i^{n-(j-1)}$, the kth loop condition after the nth iteration can be rewritten as

$$\begin{aligned} &C_{k11}\mathbf{y}(0)\,\lambda_1^n + C_{k12}\mathbf{y}(0)n\,\lambda_1^{n-1} + \cdots + C_{k1n_1}\mathbf{y}(0)\tbinom{n}{n_1-1}\lambda_1^{n-(n_1-1)} + \cdots + \\ &C_{kl1}\mathbf{y}(0)\,\lambda_l^n + C_{kl2}\mathbf{y}(0)n\,\lambda_l^{n-1} + \cdots + C_{kln_l}\mathbf{y}(0)\tbinom{n}{n_l-1}\lambda_l^{n-(n_l-1)} > 0, \end{aligned} \tag{4}$$

whose left side is denoted as $Cond_k(\mathbf{y}(n))$.

If two eigenvalues λ_i and λ_j are the same, then it is assumed that the corresponding coefficients (of $\binom{s}{t}\lambda_i^{s-t}$ and $\binom{s}{t}\lambda_j^{s-t}$) have been merged in the above expression. Therefore, it is assumed $0 < \lambda_1 < \lambda_2 < \cdots < \lambda_l$ in what follows. In [Tiw04], an order among $\binom{n}{j}\lambda_r^{n-j}$ is defined as follows: If $\lambda_i < \lambda_j$, then $\binom{n}{t_1}\lambda_i^{n-t_1} < \binom{n}{t_2}\lambda_j^{n-t_2}$ for any t_1, t_2. If $t_1 < t_2$, then $\binom{n}{t_1}\lambda_i^{n-t_1} < \binom{n}{t_2}\lambda_i^{n-t_2}$. W.r.t. the order, the biggest non-zero term of $Cond_k(\mathbf{y}(n))$ is called its *leading term* or *dominant term*, whose coefficient is called its *leading coefficient*. Obviously, the sign of $Cond_k(\mathbf{y}(n))$ is dominated by its leading term (dominant term) as n increases. For example, if $C_{kln_l}\mathbf{y}(0) \neq 0$, then the leading term (dominant term) of $Cond_k(\mathbf{y}(n))$ is $C_{kln_l}\mathbf{y}(0)\binom{n}{n_l-1}\lambda_l^{n-(n_l-1)}$ and its leading coefficient is $C_{kln_l}\mathbf{y}(0)$, and the sign of $Cond_k(\mathbf{y}(n))$ is determined by this term as n is large enough.

Thus, the third step is to non-deterministically construct a set of linear constraints which contains at most nm linear equalities and inequalities from (4) w.r.t. the order. The total number of such sets is at most n^m, where n is the number of program variables and m is the number of conditions. Then

$$\exists \mathbf{y}(0)\forall n \in \mathbb{N}. \bigwedge_{k=1}^{m} Cond_k(\mathbf{y}(n)) > 0 \tag{5}$$

is reduced to whether there exists such a set which is satisfiable.

So, we can see the procedure given in [Tiw04] depends on numerical computation, otherwise it is impossible to generate a set of linear constraints from (4). In addition, according to [Tod92], the cost for solving a set of mn linear equalities and inequalities is $O(n^3)$, so the complexity of the procedure is $O(n^3) \times n^m$, i.e., $O(n^{m+3})$, as the costs for steps 1 & 2 are so small that it can be ignored.

While, our approach consists two steps: The first step is equivalent to the first step together the second step described above. But we symbolically compute the eigenvalues and a transition matrix P of A. In fact, we only need to consider the positive eigenvalues of A and the sub-matrix of P corresponding to these positive eigenvalues; The second step is on how to symbolically determine whether P_3 terminates, which is quite different from the above third step, we will explain in detail later.

In order to implement our algorithm, we need the following notions.

Let $\mathcal{J}_\lambda = [J_1, \ldots, J_k]$ be the Jordan blocks and $\mathcal{B}_\lambda = [B_1, B_2, \ldots, B_k]$ the condition blocks related to λ. Choose the first column from each B_i for $i = 1, \ldots, k$ and form the following matrix

$$\overline{B}_\lambda = \begin{bmatrix} b_{111} & b_{211} & \ldots & b_{k11} \\ b_{121} & b_{221} & \ldots & b_{k21} \\ \vdots & \vdots & \ldots & \vdots \\ b_{1m1} & b_{2m1} & \ldots & b_{km1} \end{bmatrix}, \quad \text{and let } \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \end{bmatrix}.$$

Definition 2 Call $\mathcal{J}_\lambda = [J_1, J_2, \ldots, J_k]$ a *candidate block set* if $\overline{B}_\lambda \mathbf{u} \geq 0$ has non-zero solutions.

Intuitively, $\mathcal{J}_\lambda$ being a candidate block set implies that there must exist an eigenvector $\mathbf{v}$ related to λ such that $BA^n\mathbf{v} \geq 0$, for any $n \in \mathbb{N}$. More detailed discussion can be found later in Lemma 3.

Definition 3 Given a candidate block set $\mathcal{J}_\lambda = [J_1, J_2, \ldots, J_k]$, the subexpression of the loop condition related to λ is $\sum_{i=1}^{k} B_i J_i^n \mathbf{x}_i$. A valuation $\mathbf{a} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k]$ of $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k]$ is called *possible non-termination input* (PNI) w.r.t. $(\mathcal{B}_\lambda, \mathcal{J}_\lambda)$, if for each component of $\sum_{i=1}^{k} B_i J_i^n \mathbf{a}_i$, either it is zero or its leading coefficient is positive. For a PNI $\mathbf{a}$ w.r.t. $(\mathcal{B}_\lambda, \mathcal{J}_\lambda)$, the *maximal omissible set* w.r.t. $(\mathcal{B}_\lambda, \mathcal{J}_\lambda, \mathbf{a})$, denoted by $E(\mathcal{B}_\lambda, \mathcal{J}_\lambda, \mathbf{a})$, is the set of all the indices of the elements of $\sum_{i=1}^{k} B_i J_i^n \mathbf{a}_i$ whose leading coefficients are positive.

Given $\mathcal{J}_\lambda$ and $\mathcal{B}_\lambda$, where $\mathcal{J}_\lambda$ is a candidate block set, if $\mathbf{c}_1$ and $\mathbf{c}_2$ both are PNIs w.r.t. $(\mathcal{B}_\lambda, \mathcal{J}_\lambda)$, so is $\mathbf{c}_1 + \mathbf{c}_2$. Furthermore, $E(\mathcal{B}_\lambda, \mathcal{J}_\lambda, \mathbf{c}_i) \subseteq E(\mathcal{B}_\lambda, \mathcal{J}_\lambda, \mathbf{c}_1 + \mathbf{c}_2) = E(\mathcal{B}_\lambda, \mathcal{J}_\lambda, \mathbf{c}_1) \cup E(\mathcal{B}_\lambda, \mathcal{J}_\lambda, \mathbf{c}_2)$, for $i = 1, 2$. Thus, we can find a PNI $\mathbf{a}$ such that $E(\mathcal{B}_\lambda, \mathcal{J}_\lambda, \mathbf{b}) \subseteq E(\mathcal{B}_\lambda, \mathcal{J}_\lambda, \mathbf{a})$ for any PNI $\mathbf{b}$. $E(\mathcal{B}_\lambda, \mathcal{J}_\lambda, \mathbf{a})$ is called the maximal omissible set (MOS) w.r.t. $(\mathcal{B}_\lambda, \mathcal{J}_\lambda)$, denoted by $E(\mathcal{B}_\lambda, \mathcal{J}_\lambda)$.

For completeness, we stipulate that $E(\mathcal{B}_\lambda, \mathcal{J}_\lambda) = \emptyset$ if $\mathcal{J}_\lambda$ is not a candidate block set. Its correctness will be shown later.

Intuitively, for each element $k_0 \in E(\mathcal{B}_\lambda, \mathcal{J}_\lambda)$, if λ is the biggest positive eigenvalue, then the k_0th loop condition (i.e., the k_0th inequality) holds when n is large enough. So, the k_0th row of B can be deleted.

For convenience, for a matrix D, we denote by $\mathcal{O}(D)$ the order of D, i.e., the number of its rows, by $\mathcal{C}(D)$ the set of all its column indices, and by $\mathcal{R}(D)$ the set of all its row indices in what follows.

Now, we present our symbolic decision procedure $\texttt{DecTerm}(B, A)$ for deciding whether a linear program $\mathtt{P}_2$ represented by two matrices A and B terminates. In $\texttt{DecTerm}(B, A)$, I_B is a set variable, with initial value $\mathcal{R}(B)$, which is used to indicate which conditions are still unsatisfiable so far. While T_A with initial value $\emptyset$ is used to indicate which eigenvalues of A have been taken into account. The algorithm also calls a subroutine FMOS which computes the maximal omissible set $E(\mathcal{B}_\lambda, \mathcal{J}_\lambda)$ for a given positive eigenvalue λ of A.

Procedure $\texttt{DecTerm}(B, A)$

/ Recall that it is assumed that A and B are $n \times n$ and $m \times n$ matrices, respectively. */*

Step 1: **1.1** Find all the positive eigenvalues of A symbolically and represent them by $\lambda_1, \ldots, \lambda_l$ in increase order;

1.2 For each λ_i, $i = 1, \ldots, l$, compute Jordan block set $\mathcal{J}_{\lambda_i} = [J_{i1}, \ldots, J_{is_i}]$ by JordanBlocks;

1.3 Compute s_i linear independent eigenvectors $\mathbf{v}_{i1}, \ldots, \mathbf{v}_{is_i}$ related to λ_i, for $i = 1, \ldots, l$ by EV. Thus, we can obtain the eigenspace $\langle \mathbf{v}_{i1}, \ldots, \mathbf{v}_{is_i} \rangle$ related to λ_i, written as V_i;

1.4 Let $e_{ij} = \mathcal{O}(J_{ij}) - 1$. For $i = 1, \ldots, l$ and $j = 1, \ldots, s_i$, by GEV we can obtain e_{ij} linear independent generalized eigenvectors $\mathbf{u}_{ij1}, \ldots, \mathbf{u}_{ije_{ij}}$ related to J_{ij}. Let

$$P = [\mathbf{v}_{11}, \mathbf{u}_{111}, \ldots, \mathbf{u}_{11e_{11}}, \ldots, \mathbf{v}_{1s_1}, \mathbf{u}_{1s_11}, \ldots, \mathbf{u}_{1s_1e_{1s_1}}, \ldots, \mathbf{v}_{l1}, \mathbf{u}_{l11}, \ldots, \mathbf{u}_{l1e_{l1}}, \ldots, \mathbf{v}_{ls_l}, \mathbf{u}_{ls_l1}, \ldots, \mathbf{u}_{ls_le_{ls_l}}],$$

then P is a matrix such that $AP = P\texttt{Diag}(J_{11}, \ldots, J_{1s_1}, \ldots, J_{l1}, \ldots, J_{ls_l})$.[4]

1.5 According to $[J_{11}, \ldots, J_{1s_1}, \ldots, J_{l1}, \ldots, J_{ls_l}]$, partition BP into blocks $[B_{11}, \ldots, B_{1s_1}, \ldots, B_{l1}, \ldots, B_{ls_l}]$. Denote by $\mathcal{B}_{\lambda_i}$ $[B_{i1}, \ldots, B_{is_i}]$ the condition block set related to λ_i, where $i = 1, \ldots, l$.

Step 2: Set $i := l$, $T_A := \emptyset$, $I_B = \mathcal{R}(B) = \{1, \ldots, m\}$.

Step 3: **3.1** Set $j := i$. Check whether there is an eigenvector $\mathbf{v}$ in $V_1, V_2, \ldots, V_i$ s.t. $\bigwedge_{k \in I_B} \mathbf{b}_k \mathbf{v} > 0$, where $\mathbf{b}_k$ stands for the kth row of B. If $\exists 1 \le r \le i\, \exists \mathbf{v} \in V_r. \bigwedge_{k \in I_B} \mathbf{b}_k \mathbf{v} > 0$, then set $T_A := T_A \cup \{\lambda_r\}$ and $E(\mathcal{B}_{\lambda_j}, \mathcal{J}_{\lambda_j}) := I_B$, and GOTO 3.4.

3.2 Check whether there is an eigenvector $\mathbf{v}$ in V_j s.t. $B\mathbf{v} \ge 0$. If no then if $j = 1$ then $E(\mathcal{B}_{\lambda_j}, \mathcal{J}_{\lambda_j}) := \emptyset$ and GOTO 3.4; otherwise set $j := j - 1$ and GOTO 3.2;

3.3 Compute $E(\mathcal{B}_{\lambda_j}, \mathcal{J}_{\lambda_j})$ by $\texttt{FMOS}(\mathcal{B}_{\lambda_j}, \mathcal{J}_{\lambda_j}, \lambda_j)$, which will be given later;

3.4 1. If $E(\mathcal{B}_{\lambda_j}, \mathcal{J}_{\lambda_j}) = I_B$ then RETURN non-terminating;

2. If $j = 1$ then RETURN terminating; otherwise,

(a) Say $E(\mathcal{B}_{\lambda_j}, \mathcal{J}_{\lambda_j}) = \{c_1, \ldots, c_r\}$, where $r \le \mid I_B \mid$. $I_B := I_B - E(\mathcal{B}_{\lambda_j}, \mathcal{J}_{\lambda_j})$ (remove $c_1, \ldots, c_r$ from I_B) and $i := j - 1$;

(b) GOTO 3.1.

Step 1 is to symbolically reduce the termination problem of $\mathtt{P}_2$ to that of $\mathtt{P}_4$ via $\mathtt{P}_3$; while **Step 2** is quite simple, just to initialize the variables. Note that I_B is a global variable, will be used in the subroutine FMOS. **Step 3** is to symbolically determine whether $\mathtt{P}_4$ terminates. The basic idea is: Firstly, according to the result given in the previous section, we calculate m conditions also with form (4), but in which all expressions are symbolic. Note that in our case the coefficients of $\binom{n}{j} \lambda_r^{n-j}$ in (4) may contain $\lambda_1, \ldots, \lambda_l$ too, but their degrees are fixed and independent of n, and the order among $\binom{n}{j} \lambda_r^{n-j}$ still holds; Secondly, in order to check if (5) holds, we consider these positive eigenvalues in turn from the biggest one to the smallest one. Consider λ_h, where $1 \le h \le l$. If there is an eigenvector $\mathbf{v}$ related to some λ_i for some $1 \le i \le l$, such that for each remained condition $k \in I_B$, $\mathbf{b}_k \mathbf{v} > 0$, then return the program does not terminate and the algorithm stops; otherwise, check whether $\mathcal{J}_h$ is a candidate block set. This is equivalent to check whether there is an eigenvector $\mathbf{v}_h \in V_h$ such that $B\mathbf{v}_h \ge 0$. If not, then go to λ_{h-1}; otherwise, we focus on the subexpressions of

[4] Since A may have non-positive eigenvalues, the resulting matrix P is just a submatrix of some transition matrix P^* of A such that $(P^*)^{-1}AP^*$ is the Jordan normal form of A. In addition, sub-steps 1.3 and 1.4 can be removed in fact as the functions of EV and GEV are included in JordanBlocks. For clarity, we keep them here.

$Cond_k$ only related to λ_h for all $k \in I_B$ and symbolically determine which conditions indexed in I_B always keep positive after iterating n times when n is large enough by solving SASs. If all remained conditions hold, then return the program is non-terminating and the algorithm stops; otherwise, update I_B by removing the indices of which conditions are positive when n is big enough and go to λ_{h-1}. The algorithm starts the above procedure from λ_l and repeat it until λ_1. Finally, if there still are some conditions remained unsatisfiable, i.e., $I_B \neq \emptyset$, then the program terminates; otherwise, the program does not terminate.

It is not hard to prove the following equation

$$B_{ij} J_{ij}^n \mathbf{x}_{ij} = \begin{bmatrix} b_{ij11} & b_{ij12} & \ldots & b_{ij1(e_{ij}+1)} \\ b_{ij21} & b_{ij22} & \ldots & b_{ij2(e_{ij}+1)} \\ \vdots & \vdots & \ldots & \vdots \\ b_{ijm1} & b_{ijm2} & \ldots & b_{ijm(e_{ij}+1)} \end{bmatrix} \begin{bmatrix} \lambda_i^n & n\,\lambda_i^{n-1} & \ldots & \binom{n}{e_{ij}} \lambda_i^{n-e_{ij}} \\ 0 & \lambda_i^n & \ldots & \binom{n}{e_{ij}-1} \lambda_i^{n-(e_{ij}-1)} \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \ldots & \lambda_i^n \end{bmatrix} \begin{bmatrix} x_{ij1} \\ x_{ij2} \\ \vdots \\ x_{ij(e_{ij}+1)} \end{bmatrix}$$

$$= B_{ij} \begin{bmatrix} x_{ij1} \\ x_{ij2} \\ \vdots \\ x_{ije_{ij}} \\ x_{ij(e_{ij}+1)} \end{bmatrix} \lambda_i^n + B_{ij} \begin{bmatrix} x_{ij2} \\ x_{ij3} \\ \vdots \\ x_{ij(e_{ij}+1)} \\ 0 \end{bmatrix} n\,\lambda_i^{n-1} + \cdots + B_{ij} \begin{bmatrix} x_{ij(e_{ij}+1)} \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \binom{n}{e_{ij}} \lambda_i^{n-e_{ij}}, \quad (6)$$

where $i = 1, \ldots, l$ and $j = 1, \ldots, s_i$, and $e_{ij} = \mathcal{O}(J_{ij}) - 1$.

We can present the procedure `FMOS`, to compute $E(\mathcal{B}_\lambda, \mathcal{J}_\lambda)$, as follows:

function `FMOS`$(\mathcal{B}_{\lambda_j}, \mathcal{J}_{\lambda_j}, \lambda_j)$

Step 1: Compute $\sum_{k=1}^{s_j} B_{jk} J_{jk}^n \mathbf{x}_{jk}$ and collect all the coefficients of $\binom{n}{f} \lambda_j^{n-f}$. Then the kth component can be represented by $Cond_{kj} = C_{kj1}\,\lambda_j^n + C_{kj2}\, n\,\lambda_j^{n-1} + \cdots + C_{kje_j} \binom{n}{e_j-1} \lambda_j^{n-(e_j-1)}$, for $k \in I_B$, where $e_j = \max\{\mathcal{O}(J_{jt}) \mid 1 \le t \le s_j\}$;

Step 2:

FOR $I = |I_B|$ DOWN TO 1 DO
  FOR each $\{c_1, \ldots, c_i\} \in Q_i$ DO
    FOR each $(d_1, \ldots, d_i) \in \{1, \ldots, e_j\}^i$ DO

1. Construct an SAS as follows:

$$\begin{cases} C_{c_r jh} = 0 & \text{for } d_r < h \le e_j, \quad r = 1, \ldots, i, \\ C_{e_r j d_r} > 0, & \text{for } r = 1, \ldots, i, \\ C_{ejd} = 0, & \text{for } e \in I_B \setminus \{c_1, \ldots, c_i\}, \quad d = 1, \ldots, e_j. \end{cases}$$

2. If the above SAS has solutions, then
   (a) Set $E(\mathcal{B}_{\lambda_j}, \mathcal{J}_{\lambda_j}) := \{c_1, \ldots, c_i\}$;
   (b) RETURN $E(\mathcal{B}_{\lambda_j}, \mathcal{J}_{\lambda_j})$.

where $Q_i = \{\{c_1, \ldots, c_i\} \mid \{c_1, \ldots, c_i\} \subseteq I_B\}$.

3.3. Example

We use the following example to illustrate the procedure of `DecTerm`.

`Q3` **while** $(B\mathbf{x} > 0)$ $\{\mathbf{x} := A\mathbf{x}\}$, where $A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 3 & 0 & 0 & -3 & 0 \\ 0 & 1 & 0 & 3 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ and $B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$.

Step 1: **1.1** The characteristic polynomial of A is $D(\lambda) = (\lambda - 1)(\lambda^2 - 3)^2$. We know that A has two positive eigenvalues, i.e., $\lambda_1 = 1$ with multiplicity 1 and λ_2 which is the positive root of $\lambda^2 - 3$ with multiplicity 2. By DISCOVERER, a possible interval containing λ_2 is $[1, 2]$.[5]

1.2 Let's see how `JordanBlocks` works. From Example 2, we know that $rank(A - \lambda_2 I) = 4$. By `FFGE`, we can find that $rank((A - \lambda_2 I)^2) = 3$ and $rank((A - \lambda_2 I)^3) = 3$. Then, A has a Jordan block related to λ_2 with order 2. Since the multiplicity of λ_2 is 2, we have $\mathcal{J}_{\lambda_2} = [J_{21}]$, where $J_{21} = \begin{bmatrix} \lambda_2 & 1 \\ 0 & \lambda_2 \end{bmatrix}$. Since the multiplicity of λ_1 is 1, $\mathcal{J}_{\lambda_1} = [J_{11}]$, where $J_{11} = [\lambda_1]$.

1.3 Compute a generalized eigenvector of rank $k = 2$ as follows: By `FFGE`, solve $(A - \lambda_2 I)^2\mathbf{v} = 0$ subject to $(A - \lambda_2 I)\mathbf{v} \neq 0$ and obtain $\mathbf{v}_{22} = [2, 2\lambda_2, 1, 0, 0]^T$. So, the related eigenvector is $\mathbf{v}_{21} = (A - \lambda_2 I)\mathbf{v}_{22} = [1, 0, \lambda_2, 1, 0]^T$. Since there is only one Jordan block related to λ_2, the eigenspace related to λ_2 is $V_2 = \langle \mathbf{v}_{21} \rangle$. It's easy to find that the eigenvector related to λ_1 is $\mathbf{v}_{11} = [0, 0, 0, 0, 1]^T$ and the related eigenspace is $V_1 = \langle \mathbf{v}_{11} \rangle$.

1.4 Let $e_{11} = \mathcal{O}(J_{11}) - 1 = 0$ and $e_{21} = \mathcal{O}(J_{21}) - 1 = 1$. One generalized eigenvector related to λ_2 is computed in Step 1.3, so $P = [v_{11}, v_{21}, v_{22}]$. Accordingly, $BP = \begin{bmatrix} 0 & 1 & 2 \\ 0 & 0 & 2\lambda_2 \\ 0 & \lambda_2 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$.

1.5 According to $[J_{11}, J_{21}]$, we have $B_{11} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ and $B_{21} = \begin{bmatrix} 1 & 2 \\ 0 & 2\lambda_2 \\ \lambda_2 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}$. Thus, $\mathcal{B}_{\lambda_1} = [B_{11}]$ and $\mathcal{B}_{\lambda_2} = [B_{21}]$.

Step 2: Set $i := 2$, $T_A := \emptyset$, and $I_B := \mathcal{R}(B) = \{1, 2, 3, 4, 5\}$.

Step 3: **3.1** Set $j := 2$. We can find that there is no eigenvector $\mathbf{v}$ in V_1 and V_2 s.t. $B\mathbf{v} > 0$.

3.2 We can find that $B\mathbf{v}_{21} \geq 0$, therefore $T_A = T_A \bigcup \{\lambda_2\}$.

3.3 By calling `FMOS`$(\mathcal{B}_{\lambda_2}, \mathcal{J}_{\lambda_2}, \lambda_2)$, we get $E(\mathcal{B}_{\lambda_2}, \mathcal{J}_{\lambda_2}) = \{1, 2, 3, 4\}$.

3.4 Since $E(\mathcal{B}_{\lambda_2}, \mathcal{J}_{\lambda_2}) \neq I_B$ and $j \neq 1$, $I_B := I_B - E_{\lambda_2} = \{5\}$.

3.1 Since $B\mathbf{v}_{11} > 0$, $T_A := T_A \bigcup \{\lambda_1\}$ and $E(\mathcal{B}_{\lambda_1}, \mathcal{J}_{\lambda_1}) = I_B = \{5\}$.

3.4 Since $E(\mathcal{B}_{\lambda_1}, \mathcal{J}_{\lambda_1}) = I_B$, return non-terminating.

4. Correctness

In this section, we will prove the correctness of the algorithm `DecTerm`(B, A).

Lemma 1 *Assume $\mathcal{J}_\lambda = [J_1, J_2, \ldots, J_t]$ and $\mathcal{B}_\lambda = [B_1, B_2, \ldots, B_t]$ are the Jordan block set and condition block set related to eigenvalue λ, respectively. The loop*

P_5 **while** $(\mathcal{B}_\lambda \mathbf{x} > 0)\{\mathbf{x} := Diag(\mathcal{J}_\lambda)\mathbf{x}\}$

does not terminate if and only if $\mathcal{J}_\lambda$ is a candidate block set and $E(\mathcal{B}_\lambda, \mathcal{J}_\lambda) = \{1, 2, \ldots, \mathcal{O}(B_1)\}$.

Proof. $\Rightarrow$ Let $s_1 = \mathcal{O}(J_1), \ldots, s_t = \mathcal{O}(J_t)$. Without loss of generality, assume $s_1 < s_2 < \cdots < s_{t-1} = s_t$. Since the loop is non-terminating, there exists $\mathbf{y} = [\mathbf{y}_1, \ldots, \mathbf{y}_t]^T$ s.t. the leading coefficient of $Cond_k(\mathbf{y}(n))$ is positive

[5] Using DISCOVERER, one can get an interval containing λ_2 with arbitrarily small length.

for $k \in \{1, \dots, \mathcal{O}(B_1)\}$, where $\mathbf{y}_1 = [y_{11}, \dots, y_{1s_1}]^T$, $\mathbf{y}_2 = [y_{21}, \dots, y_{2s_2}]^T, \dots, \mathbf{y}_t = [y_{t1}, \dots, y_{ts_t}]^T$. According to (6), we expand $B_1 J_1^n \mathbf{y}_1 + B_2 J_2^n \mathbf{y}_2 + \cdots + B_t J_t^n \mathbf{y}_t$ and denote the resulting expression by $f(\lambda)$.

Suppose $\mathcal{J}_\lambda$ is not a candidate block set, that is

$$\begin{bmatrix} b_{111} & b_{211} & \dots & b_{t11} \\ b_{121} & b_{221} & \dots & b_{t21} \\ \vdots & \vdots & \dots & \vdots \\ b_{1l1} & b_{2l1} & \dots & b_{tl1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{bmatrix} \geq 0 \tag{7}$$

has no non-zero solutions by Definition 2.

Consider first $\binom{n}{s_t-1} \lambda^{n-(s_t-1)}$ that is the biggest among of $\binom{n}{j} \lambda^{n-j}$ w.r.t. the order. Its coefficient must be greater than or equal to 0, otherwise, it is easy to see that $\mathtt{P}_5$ terminates, i.e.,

$$B_{t-1} \begin{bmatrix} y_{(t-1)s_{t-1}} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + B_t \begin{bmatrix} y_{ts_t} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} b_{(t-1)11} & b_{t11} \\ b_{(t-1)21} & b_{t21} \\ \vdots & \vdots \\ b_{(t-1)l1} & b_{tl1} \end{bmatrix} \begin{bmatrix} y_{(t-1)s_{t-1}} \\ y_{ts_t} \end{bmatrix} \geq 0. \tag{8}$$

By (7), (8) has no non-zero solution, otherwise it is easy to construct a non-zero solution for (7). Thus, $y_{(t-1)s_{t-1}} = y_{ts_t} = 0$.

Now, let us consider $\binom{n}{s_t-2} \lambda^{n-(s_t-2)}$ in $f(\lambda)$. Similarly, its coefficient is no less than 0,[6] i.e.,

$$B_{t-1} \begin{bmatrix} y_{(t-1)s_{t-1}-1} \\ y_{(t-1)s_{t-1}} \\ \vdots \\ 0 \end{bmatrix} + B_t \begin{bmatrix} y_{ts_t-1} \\ y_{ts_t} \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} b_{(t-1)11} & b_{t11} \\ b_{(t-1)21} & b_{t21} \\ \vdots & \vdots \\ b_{(t-1)l1} & b_{tl1} \end{bmatrix} \begin{bmatrix} y_{(t-1)s_{t-1}-1} \\ y_{ts_t-1} \end{bmatrix} \geq 0. \tag{9}$$

Similarly, it results that $y_{(t-1)s_{t-1}-1} = y_{ts_t-1} = 0$ as (7) has no non-zero solution.

Repeat the procedure, we can obtain $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_t] = [\overbrace{0, 0, \dots, 0}^{s_1+\cdots+s_t}]$, which contradicts to that the leading coefficient of $Cond_k(\mathbf{y}(n))(k \in \{1, \dots, \mathcal{O}(B_1)\})$ is positive. This concludes that $\mathcal{J}_\lambda = [J_1, J_2, \dots, J_m]$ is a candidate block set.

If $E(B_\lambda, J_\lambda) \neq \{1, \dots, \mathcal{O}(B_1)\}$, then for any input $\mathbf{c}$, the omissible set of $\mathbf{c}$ is a proper subset of $E(B_\lambda, J_\lambda)$, so $\mathbf{c}$ must make the loop terminate, which contradicts to $\mathtt{P}_5$ does not terminate. Thus, $E(B_\lambda, J_\lambda) = \{1, \dots, \mathcal{O}(B_1)\}$.

$\Leftarrow$ If $\mathcal{J}_\lambda$ is a candidate block set and $E(\mathcal{B}_\lambda, \mathcal{J}_\lambda) = \{1, 2, \dots, \mathcal{O}(B_1)\}$, then there exists a PNI $\mathbf{c}_0$ such that $E(\mathcal{B}_\lambda, \mathcal{J}_\lambda, A^i\mathbf{c}_0) = E(\mathcal{B}_\lambda, \mathcal{J}_\lambda)$ when i is large enough according to Definition 3. Consequently, the loop is therefore non-terminating on input $A^i\mathbf{c}_0$. □

Remark 1 Similar to the proof of the if part of Lemma 1, we can prove that if $\mathcal{J}_\lambda$ is not a candidate block set, then in order to ensure that all the leading coefficients of $\mathcal{B}_\lambda \mathcal{J}_\lambda^n \mathbf{z}$ are non-negative, $\mathbf{z}$ must be zero. This guarantees the correctness of the definition of $E(\mathcal{B}_\lambda, \mathcal{J}_\lambda)$ in the case when $\mathcal{J}_\lambda$ is not a candidate block set. We shall prove later that $\mathcal{J}_\lambda = [J_1, J_2, \dots, J_t]$ is a candidate block set if and only if there is an eigenvector $\mathbf{v}$ related to λ such that $B\mathbf{v} \geq 0$.

Lemma 2 *If there is an eigenvector* $\mathbf{v}$ *related to a positive eigenvalue* λ *s.t.* $B\mathbf{v} > 0$*, then* $E(\mathcal{B}_\lambda, \mathcal{J}_\lambda) = \mathcal{R}(B)$.

Proof. Let $\mathbf{v}$ be an eigenvector, $\mathcal{J}_\lambda$ the Jordan block set and $\mathcal{B}_\lambda$ the condition block set related to λ. $J_i \in \mathcal{J}_\lambda$, $B_i \in \mathcal{B}_\lambda$. Suppose $\mathbf{v}^1, \dots, \mathbf{v}^j$ are the generalized eigenvectors related to J_i. Since $B_i = [B\mathbf{v}, B\mathbf{v}^1, \dots, B\mathbf{v}^j]$, it follows that each component of the first column of B_i is positive. Let $\mathbf{c}_i = [1, \overbrace{0, \dots, 0}^{\mathcal{O}(J_i)-1}]^{\mathrm{T}}$ and $\mathbf{c}_r = [\overbrace{0, \dots, 0}^{\mathcal{O}(J_r)}]^{\mathrm{T}}$ for $1 \leq r \leq t \wedge r \neq i$. Thus, $B_1 J_1^n \mathbf{c}_1 + B_2 J_2^n \mathbf{c}_2 + \cdots + B_t J_t^n \mathbf{c}_t = B_i J_i^n \mathbf{c}_i$. It's easy to see that the leading

[6] For simplicity, we here only consider the case when $s_{t-2} < s_{t-1}$. For the case $s_{t-2} = s_{t-1}$, it can be handled similarly.

coefficient of each loop condition is positive, and the omissible set of $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_i, \ldots, \mathbf{c}_t)$ consists of all the row indices of B. Thus, $E(\mathcal{B}_\lambda, \mathcal{J}_\lambda) = \mathcal{R}(B)$ from Definition 3. □

Remark 2 The above lemma ensures the correctness of **Step 3.1**.

The following lemma indicates the relation between a candidate block set and the eigenvectors for a given positive eigenvalue.

Lemma 3 *Given a positive eigenvalue λ, $\mathcal{J}_\lambda = [J_1, J_2, \ldots, J_t]$ is a candidate block set if and only if there is an eigenvector $\mathbf{v}$ related to λ such that $B\mathbf{v} \geq 0$.*

Proof. $\Rightarrow$ Let $s_1, \ldots, s_t$ be the orders of $J_1, \ldots, J_t$ respectively, and $\mathbf{y} = [y_1, y_2, \ldots, y_t]$ a non-zero vector over $\mathbb{R}$ such that $\begin{bmatrix} b_{111} & b_{211} & \ldots & b_{t11} \\ b_{121} & b_{221} & \ldots & b_{t21} \\ \vdots & \vdots & \ldots & \vdots \\ b_{1l1} & b_{2l1} & \ldots & b_{tl1} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_t \end{bmatrix} \geq 0$. Let

$$\mathbf{v}_1 = [1, \overbrace{0, \ldots, 0}^{s_1+\cdots+s_t-1}]^T, \mathbf{v}_2 = [\overbrace{0, \ldots, 0}^{s_1}, 1, \overbrace{0, \ldots, 0}^{s_2+\cdots+s_t-1}]^T, \ldots, \mathbf{v}_t = [\overbrace{0, \ldots, 0}^{s_1+\cdots+s_{t-1}}, 1, \overbrace{0, \ldots, 0}^{s_t-1}]^T$$

that are the standard orthogonal basis of the eigenspace related to λ. Let $\mathbf{v} = y_1\mathbf{v}_1 + y_2\mathbf{v}_2 + \cdots + y_t\mathbf{v}_t$. It gives

$$[B_1, B_2, \ldots, B_t]\mathbf{v} = \begin{bmatrix} b_{111} & b_{211} & \ldots & b_{t11} \\ b_{121} & b_{221} & \ldots & b_{t21} \\ \vdots & \vdots & \ldots & \vdots \\ b_{1l1} & b_{2l1} & \ldots & b_{tl1} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_t \end{bmatrix} \geq 0. \tag{10}$$

Obviously, $\mathbf{v}$ is a non-zero eigenvector satisfying $B\mathbf{v} \geq 0$.

$\Leftarrow$ Let $\mathbf{v}_1, \ldots, \mathbf{v}_t$ be the eigenvectors corresponding to $J_1, \ldots, J_t$ respectively. Suppose $\mathbf{v}$ is an eigenvector related to λ such that $B\mathbf{v} \geq 0$, therefore there exist $y_1, \ldots, y_t$ such that $\mathbf{v} = y_1\mathbf{v}_1 + \cdots + y_t\mathbf{v}_t$ and $\mid y_1 \mid + \cdots + \mid y_t \mid \neq 0$. It follows that $\mathbf{y} = [y_1, \ldots, y_t]$ is a non-zero solution of (10). This ensures $\mathcal{J}_\lambda$ is a candidate block set according to Definition 2. □

Theorem 1 *If* $\mathtt{P}_2$ *is non-terminating, then there must be a positive eigenvalue λ such that $\mathcal{J}_\lambda = [J_1, J_2, \ldots, J_t]$ is a candidate block set.*

Proof. For brevity, assume that A has two distinct positive eigenvalues $0 < \lambda_1 < \lambda_2$. The other cases can be proved similarly. Therefore, let $\mathcal{J}_{\lambda_1} = [J_{11}, J_{12}, \ldots, J_{1t_1}]$ and $\mathcal{J}_{\lambda_2} = [J_{21}, J_{22}, \ldots, J_{2t_2}]$ be the Jordan block sets related to λ_1 and λ_2 respectively. Then $BA^n\mathbf{x} = (B_{11}J_{11}^n\mathbf{x}_{11} + \cdots + B_{1t_1}J_{1t_1}^n\mathbf{x}_{1t_1}) + (B_{21}J_{21}^n\mathbf{x}_{21} + \cdots + B_{2t_2}J_{2t_2}^n\mathbf{x}_{2t_2})$. Because $\mathtt{P}_2$ is non-terminating, there exists a PNI $\mathbf{y} = [\mathbf{y}_{11}, \ldots, \mathbf{y}_{1t_1}, \mathbf{y}_{21}, \ldots, \mathbf{y}_{2t_2}]^T$ such that the leading coefficient of $Cond_k(\mathbf{y}(n))$ for $k = 1, \ldots, \mathcal{O}(B)$ is positive. If $\mathcal{J}_{\lambda_2} = [J_{21}, J_{22}, \ldots, J_{2t_2}]$ is not a candidate block set, then $E(\mathcal{B}_{\lambda_2}, \mathcal{J}_{\lambda_2}) = \emptyset$. It follows $\mathbf{y}_{21} = \cdots = \mathbf{y}_{2t_2} = 0$, otherwise there must be some loop condition whose leading coefficient is negative. Thus, $BA^n\mathbf{y} = B_{11}J_{11}^n\mathbf{y}_{11} + \cdots + B_{1m_1}J_{1m_1}^n\mathbf{y}_{1t_1}$ for any $n > 0$. This means that

while $(\mathcal{B}_{\lambda_1}\mathbf{x} > 0)$ $\{\mathbf{x} := Diag(\mathcal{J}_{\lambda_1})\mathbf{x}\}$

does not terminate. According to Lemma 1, $\mathcal{J}_{\lambda_1}$ is a candidate block set. □

From Lemma 3 and Theorem 1, it is easy to prove the following result which was first proved in [Tiw04].

Corollary 1 *If the program* $\mathtt{P}_2$ *is non-terminating, then there must be a real eigenvector $\mathbf{v}$ of A corresponding to a positive eigenvalue such that $B\mathbf{v} \geq 0$.*

A direct corollary of Corollary 1 is

Corollary 2 *Assume that for every real eigenvector $\mathbf{v}$ of A corresponding to a positive eigenvalue, every element of $B\mathbf{v}$ is not zero. Then, program* $\mathtt{P}_2$ *is non-terminating if and only if there is a real eigenvector $\mathbf{v}$ of A corresponding to a positive eigenvalue such that $B\mathbf{v} > 0$.*

Theorem 2 *P_2 is non-terminating if and only if $\bigcup_{\lambda \in T'_A} E(\mathcal{B}_\lambda, \mathcal{J}_\lambda) = \mathcal{R}(B)$, where T'_A is the final value of T_A when* `DecTerm` (B, A) *terminates.*

Proof. Consider the following cases:

(i) A has no positive eigenvalue;
(ii) A has only one positive eigenvalue;
(iii) A has more than one positive eigenvalues.

The theorem holds obviously in the case (i). In addition, Lemma 1 guarantees the theorem holds in the case (ii) also. In the following, we only focus on the last case. Suppose all the distinct positive eigenvalues of A are $\lambda_1 < \lambda_2 < \cdots < \lambda_l$, where $l > 1$.

$\Rightarrow$ Suppose $\mathtt{P}_2$ is non-terminating. By Theorem 1 there exists a positive eigenvalue λ such that $\mathcal{J}_\lambda = [J_1, J_2, \ldots, J_t]$ is a candidate block set. Furthermore, there exists an input $\mathbf{c}$ s.t. all the dominant terms of $Cond_k(\mathbf{c}(n))$ for $1 \leq k \leq m$ are related to the positive eigenvalues and their coefficients are positive. From the basics of Linear Algebra, the space consisting of all the eigenvectors and the generalized eigenvectors of A in $\mathbb{R}^n$ related to all its eigenvalues is the whole space $\mathbb{R}^n$. Suppose $\mathbf{v}_1, \ldots, \mathbf{v}_s, \mathbf{v}_{s+1}, \ldots, \mathbf{v}_n$ are a base of the space, where $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_s$ are related to the positive eigenvalues and $\mathbf{v}_{s+1}, \ldots, \mathbf{v}_n$ are related to the other eigenvalues of A. Therefore, $\mathbf{c} = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_s\mathbf{v}_s + c_{s+1}\mathbf{v}_{s+1} + \cdots + c_n\mathbf{v}_n$, where $c_i \in \mathbb{R}$ for $i = 1, \ldots, n$. Let $\mathbf{y} = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_s\mathbf{v}_s$. It can be shown that the dominant term of $Cond_k(\mathbf{c}(\ell))$ must be same as that of $Cond_k(\mathbf{y}(\ell))$; moreover, its coefficient must be positive, for $k \in \{1, \ldots, m\}$. Suppose $\mathbf{y} = \mathbf{y}_1 + \mathbf{y}_2 + \cdots + \mathbf{y}_l$, where $\mathbf{y}_i$ is a non-zero linear combination of the components of $[\mathbf{v}_1, \ldots, \mathbf{v}_s]$ related to λ_i, for $i = 1, \ldots, l$, where $l \leq s \leq n$.

Firstly, compute $E(\mathcal{B}_{\lambda_l}, \mathcal{J}_{\lambda_l})$ that contains $E(\mathcal{B}_{\lambda_l}, \mathcal{J}_{\lambda_l}, \mathbf{y}_l)$ and then delete the rows of B whose indices are contained in $E(\mathcal{B}_{\lambda_l}, \mathcal{J}_{\lambda_l})$; Subsequently, repeat the above procedure until λ_1. Since all the leading coefficients of $Cond_k(\mathbf{y}(\ell))$ for $1 \leq k \leq m$ are positive, $\bigcup_{i=1}^{l} E(\mathcal{B}_{\lambda_i}, \mathcal{J}_{\lambda_i}, \mathbf{y}_i) = \mathcal{R}(B)$, so $\bigcup_{i=1}^{l} E(\mathcal{B}_{\lambda_i}, \mathcal{J}_{\lambda_i}) = \mathcal{R}(B)$ according to Definition 3.

On the other hand, by induction on the execution of `DecTerm` (mainly on **Step 3**), we have

$$\bigcup_{\lambda \in T_A} E(\mathcal{B}_\lambda, \mathcal{J}_\lambda) \supseteq \bigcup_{\lambda' \geq \min(T_A)} E(\mathcal{B}_{\lambda'}, \mathcal{J}_{\lambda'}) \tag{11}$$

$$\bigcup_{\lambda \in T_A} E(\mathcal{B}_\lambda, \mathcal{J}_\lambda) \cup I_B = \mathcal{R}(B). \tag{12}$$

Suppose $\bigcup_{\lambda \in T'_A} E(\mathcal{B}_\lambda, \mathcal{J}_\lambda) \neq \mathcal{R}(B)$. According to (12), we have $E(\mathcal{B}_{\lambda_1}, \mathcal{J}_{\lambda_1}) \subset I_B$, that is,

$$\bigcup_{\lambda \in T'_A} E(\mathcal{B}_\lambda, \mathcal{J}_\lambda) \subset \mathcal{R}(B). \tag{13}$$

However, by (11), we have

$$\bigcup_{\lambda \in T'_A} E(\mathcal{B}_\lambda, \mathcal{J}_\lambda) \supseteq \bigcup_{\lambda' \geq \lambda_1} E(\mathcal{B}_{\lambda'}, \mathcal{J}_{\lambda'}) \supseteq \bigcup_{i=1}^{l} E(\mathcal{B}_{\lambda_i}, \mathcal{J}_{\lambda_i}) = \mathcal{R}(B) \tag{14}$$

(13) contradicts to (14), therefore, $\bigcup_{\lambda \in T'_A} E(\mathcal{B}_\lambda, \mathcal{J}_\lambda) = \mathcal{R}(B)$.

($\Leftarrow$) Suppose $\bigcup_{\lambda \in T'_A} E(\mathcal{B}_\lambda, \mathcal{J}_\lambda) = \mathcal{R}(B)$. For each $\lambda_{T_i} \in T'_A$, where $1 \leq i \leq h = |T'_A|$, according to Definition 3, there exists a vector $\mathbf{c}_i$ such that $E(\mathcal{B}_{\lambda_i}, \mathcal{J}_{\lambda_i}) = E(\mathcal{B}_{\lambda_i}, \mathcal{J}_{\lambda_i}, \mathbf{c}_i)$. Without loss of generality, assume $\lambda_{T_1} < \cdots < \lambda_{T_h}$. For $1 \leq i \leq l$, if there exists $1 \leq j < h$ such that $\lambda_i = \lambda_{T_j}$, then $\mathbf{b}_i = \mathbf{c}_j$; otherwise, let $\mathbf{b}_i = [\overbrace{0, 0, \ldots, 0, 0}^{\mathcal{O}(Diag(\mathcal{J}_{\lambda_i}))}]$. Let $\mathbf{b} = [\mathbf{b}_1^{\mathrm{T}}, \ldots, \mathbf{b}_l^{\mathrm{T}}]^{\mathrm{T}}$. It can be proved that all the leading terms of $Cond_k(\mathbf{b}(\ell))$ are related to the positive eigenvalues and their coefficients are positive. Consequently, the loop is non-terminating. $\square$

According to the above results, we can claim

Theorem 3 (Correctness) *The algorithm* `DecTerm` *is* correct.

Proof. First of all, since each procedure (function) called in `DecTerm` terminates, thus the first two steps terminate and the third step terminates in at most n steps. That is, `DecTerm` itself terminates.

Soundness and *completeness* of `DecTerm` is guaranteed by Theorem 2. □

5. Complexity

Recall that $n = \mathcal{O}(A)$ and $m = \mathcal{O}(B)$.

The complexity analysis for 1.1 in Step 1 is as follows. Let $D(\lambda) = f_1(\lambda)^{i_1} \ldots f_k(\lambda)^{i_k}$ be the characteristic polynomial of A, where f_j is an irreducible polynomial over $\mathbb{Q}$ of degree d_j for $j = 1, \ldots, k$, and $d_1 i_1 + \cdots + d_k i_k = n$ where $i_j > 0$. We can first isolate all the real roots of $D(\lambda)$, i.e., compute a sequence of disjoint intervals with rational endpoints such that each interval contains one and only one real root of $D(\lambda)$. Using the well known result on the computing time for root-isolation [CoL83, Joh98], we know that the computing time for isolating the roots of $D(\lambda)$ is bounded by $O(n^6)$ if we assume that the cost for an arithmetic operation on rational numbers is a time unit.[7] Furthermore, for a given interval $[a_i, b_i]$ that contains one root of $D(\lambda)$, λ_i, the cost for computing the multiplicity of λ_i is $O(n^5)$. Thus, the total cost of 1.1 is $O(n^6)$.

The complexity analysis for 1.2 in Step 1 is as follows. Let λ_j be a positive root of f_j. Consider the cost for `JordanBlocks`. Since the multiplicity of λ_j is i_j, according to the algorithm in [MiM82] it needs to compute $rank(A-\lambda_j I)^q$, where $q \geq 1$, at most $2i_j$ times in order to obtain the Jordan blocks of A related to λ_j. To compute $rank(A - \lambda_j I)^q$, we need to take at most $n(n-1)/2$ row transformations with `FFGE` on $(A - \lambda_j I)^q$ modulated by $f_j(\lambda_j) = 0$ to triangularize $(A - \lambda_j I)^q$, where each row transformation costs at most $2d_j^2 n$ multiplications. So, the cost for computing the rank of $(A - \lambda_j I)^q$ $(1 \leq q \leq 2i_j)$ is $O(n^3 d_j^2 i_j)$. Finally, the cost for computing the Jordan blocks of A is $\sum_{j=1}^{k} O(n^3 d_j^2 i_j) \leq O(n^6)$.

The complexity analysis for 1.3 and 1.4 in Step 1 is as follows. We can analyze 1.3 and 1.4 at the same time because an eigenvector can be seen as a generalized eigenvector of rank 1. Since the Jordan blocks related to the positive eigenvalues are computed in 1.2, let us assume that the number of the Jordan blocks related to some positive eigenvalue λ_j with order $\overline{o}_{j1}, \ldots, \overline{o}_{jw_j}$ are $\overline{n}_{j1}, \ldots, \overline{n}_{jw_j}$ respectively, where $\sum_{h=1}^{w_j} \overline{o}_{jh} \overline{n}_{jh} = i_j$. Then we need to compute $\overline{n}_{jh}$ generalized eigenvectors of rank $\overline{o}_{jh}$ of A related to λ_j by solving $(A - \lambda_j I)^{\overline{o}_{jh}} \mathbf{x} = 0$ subject to $(A - \lambda_j I)^{\overline{o}_{jh}-1} \mathbf{x} \neq 0$ for $(1 \leq h \leq w_j)$, which are linearly independent.[8] Its complexity is $O(w_j n^3 d_j^2)$. If we have had the generalized eigenvectors $\overline{\mathbf{v}}_{i1}, \ldots, \overline{\mathbf{v}}_{i\overline{n}_{ji}}$ whose ranks are $\overline{o}_{ji}$ for $1 \leq i \leq w_j$, then we can get all the generalized eigenvectors related to λ_j like this:

$$\mathbf{v}_j = [(A - \lambda_j I)^{g_{11}} \overline{\mathbf{v}}_{11}, \ldots, (A - \lambda_j I)^{g_{1\overline{n}_{j1}}} \overline{\mathbf{v}}_{1\overline{n}_{j1}}, \ldots, (A - \lambda_j I)^{g_{w_j 1}} \overline{\mathbf{v}}_{w_j 1}, \ldots, (A - \lambda_j I)^{g_{w_j \overline{n}_{jw_j}}} \overline{\mathbf{v}}_{w_j \overline{n}_{jw_j}}],$$

where $(0 \leq g_{ef_e} \leq (\overline{o}_{je} - 1))$, for $1 \leq e \leq w_j$ and $1 \leq f_e \leq \overline{n}_{je}$. Its complexity is $O(\sum_{h=1}^{w_j} n^3 d_j^2 \overline{n}_{jh} (\overline{o}_{jh} - 1) \leq n^3 d_j^2 i_j)$. Thus, the total complexity for 1.3 and 1.4 is $\sum_{j=1}^{k} O(w_j n^3 d_j^2) + O(n^3 d_j^2 i_j) \leq O(n^6)$.

The complexity analysis for Step 1 is as follows. Since the complexity of 1.5 is linear, the total complexity for Step 1 is $O(n^6)$.

---

[7] Generally, the quantity should be multiplied by a factor $L(d)^2$ where d is the sum of the absolute values of $D(\lambda)$'s coefficients and $L(d)$ is the number of digits of d.

[8] In fact they must exist because of the Jordan blocks' structure of A.

Before analyzing the complexity of Step 3, we point out that the algorithm proposed in [Tod92] for solving constant linear system still works for linear system with coefficients in $\mathbb{Q}[\lambda_0]$. This is because all the computation in the algorithm are basic algebraic operations and the loop conditions in the algorithm can be re-formulated by determining whether the following SAS has real solutions

$$\{f(\lambda_0) = 0, g(\lambda_0) > 0, a \le \lambda_0 \le b\}, \tag{15}$$

where a, b are rational numbers and f, g are some polynomials of degrees d_j and $d_j - 1$, respectively. We can check whether the above SAS is satisfiable as follows. First, construct *Sturm–Tarski sequence* of f w.r.t. g with Euclidean division

$$\begin{aligned} G_0(x) &= f,\ G_1(x) = \mathrm{rem}(f'g(x), f(x)),\\ G_2(x) &= -\mathrm{rem}(G_0(x), G_1(x)),\\ &\ldots,\\ G_{k+1}(x) &= -\mathrm{rem}(G_{k-1}(x), G_k(x)),\\ &\ldots,\\ G_{s+1}(x) &= -\mathrm{rem}(G_{s-1}(x), G_s(x)) = 0, \end{aligned}$$

where s is less than or equal to d_j. Then, by Sturm–Tarski's theorem, if $V(f, f'g;\ a) - V(f, f'g;\ b) = 1$, then (15) is satisfiable; otherwise, it is unsatisfiable, where $V(f, f'g;\ r)$ stands for the number of sign changes in the sequence $G_0(r), G_1(r), \ldots, G_s(r)$. Since the complexity of computing $\mathrm{rem}(G_{k-1}(x), G_k(x))$ is $O(d_j^2)$, the complexity of constructing the Sturm–Tarski sequence is $O(d_j^3)$. The complexity of computing $V(f, f'g;\ r)$ is $O(d_j^2)$. It follows that the complexity of checking whether the above SAS is satisfiable is $O(d_j^3 + d_j^2) = O(d_j^3)$. Because the coefficients are polynomials in $\mathbb{Q}[\lambda_0]$ with degrees less than d_j, the complexity of multiplication of the coefficients should be multiplied by a factor d_j^2 at most. Thus, the complexity for symbolically solving linear system in our problem is $O(i_j^3(d_j^2 + d_j^3)) = O(i_j^3 d_j^3)$.[9]

The complexity analysis for 3.1 and 3.2 in Step 3 is as follows. We still use the notations in the complexity analysis for 1.3 and 1.4 in Step 1. It is not difficult to see that the complexity of 3.1 and 3.2 are the same because they need to symbolically check whether some linear SASs is satisfiable and the two linear SAS are of the same number of variables. In fact the number of variables are the dimension of V_j which is $\sum_{h=1}^{w_j} \overline{n}_{jh}$. According to the above discussion, we know that the complexity for symbolically checking whether a linear SAS containing λ is satisfiable is $O(n_v^3 d_\lambda^3)$, where n_v is the number of the variables and d_λ is the degree of the minimal polynomial of λ. It follows that the total complexity of 3.1 and 3.2 are $\sum_{j\ge1} O(2(\sum_{h=1}^{w_j} \overline{n}_{jh})^3 d_j^3) \le \sum_{j\ge1} O(i_j^3 d_j^3) \le O(n^3)$.

The complexity analysis for 3.3 and 3.4 in Step 3 is as follows. First let us analyze the complexity of 3.3 which is, in fact, the complexity of `FMOS`. Obviously, we only need to analyze the complexity of Step 2 in `FMOS`. Let $r_{\lambda_j} = \max\{O(J_{jk}) \mid J_{jk} \in \mathcal{J}_{\lambda_j}\}$ and l_{λ_j} is the order of $\mathcal{B}_{\lambda_j}$. In the worst case, we need to solve $r_{\lambda_j}^{l_{\lambda_j}} + \binom{l_{\lambda_j}}{1} r_{\lambda_j}^{l_{\lambda_j}-1} + \cdots + \binom{l_{\lambda_j}}{l_{\lambda_j}} = (r_{\lambda_j}+1)^{l_{\lambda_j}}$ linear SASs, each of which contains at most i_j variables with $\sum_{j\ge1} i_j \le n$ and a symbol, λ_j, representing a specific eigenvalue. According to the above discussion, we know that the complexity for symbolically checking whether a linear SAS containing λ_j is satisfiable is $O(i_j^3 d_j^3)$, so the total complexity of Step 2 in `FMOS` is at most $O((r_{\lambda_j}+1)^{l_{\lambda_j}} \times i_j^3 d_j^3)$. The loop in Step 3 is executed at most k times and therefore the total complexity of Step 3 is $\sum_{j=1}^{k} O((r_{\lambda_j}+1)^{l_{\lambda_j}} \times i_j^3 d_j^3) \le O(n^{m+3})$. Furthermore, if $k \gg 1$, then $\sum_{j=1}^{k}(r_{\lambda_j}+1)^{l_{\lambda_j}} \times i_j^3 d_j^3 \ll n^{m+3}$. Since the complexity of 3.4 is linear, the total complexity of 3.3 and 3.4 is $O(n^{m+3})$.

Finally the complexity of `DecTrem` is $\max\{O(n^6), O(n^{m+3})\}$.

[9] In [Tod92], the complexity of solving a linear programming problem is $O(n^3 L)$, where n is the number of variables and L is the number of bits in the input. In this paper, we assume unit cost for arithmetic operations, so we omit L herein and hereafter.

6. The irreducible case

In this section, we will prove a sufficient and necessary condition (Theorem 4) for the termination of $\mathtt{P}_2$ under the assumption that the characteristic polynomial of A is irreducible. Based on the theorem a very simple and efficient symbolic decision procedure for termination of $\mathtt{P}_2$ can be easily invented.

6.1. A necessary and sufficient condition

Theorem 4 *Suppose A and B are both matrices on the rational numbers $\mathbb{Q}$ and the characteristic polynomial $D(\lambda)$ of A is irreducible in $\mathbb{Q}[\lambda]$, the set of univariate polynomials with rational coefficients. The program $\mathtt{P}_2$ is non-terminating if and only if there is a real eigenvector $\mathbf{v}$ of A corresponding to a positive eigenvalue such that $B\mathbf{v} > 0$.*

Proof. The sufficiency is obvious, so we only prove the necessity. The irreducibility of the characteristic polynomial $D(\lambda)$ of A implies that the eigenvalues of A are pairwise distinct. Otherwise, set

$$D_1 = \gcd\left(D(\lambda), \frac{\mathrm{d}}{\mathrm{d}\,\lambda}D(\lambda)\right), \quad D_2 = \frac{D(\lambda)}{\gcd(D(\lambda), \frac{\mathrm{d}}{\mathrm{d}\,\lambda}D(\lambda))},$$

then $D(\lambda) = D_1 D_2$ is reducible over $\mathbb{Q}$. Here, gcd stands for the greatest common divisor.

Suppose $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of A. Set

$$A(\lambda) = A - \lambda I = \begin{bmatrix} a_{11} - \lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} - \lambda & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} - \lambda \end{bmatrix},$$

and denote the (i, j) algebraic complement minor of $A(\lambda)$ by $A_{ij}(\lambda)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n$.[10]

Obviously, $A_{11}(\lambda)$ is a non-zero polynomial in λ of degree $n-1$ because its leading monomial is $(-\lambda)^{n-1}$. Then for any eigenvalue λ_β ($\beta = 1, \ldots, n$), $A_{11}(\lambda_\beta) \neq 0$ since λ_β is a root of $D(\lambda)$ which is an irreducible polynomial of degree n. For each $\beta = 1, \ldots, n$, set

$$\mathbf{v}_\beta = [A_{11}(\lambda_\beta),\ A_{12}(\lambda_\beta), \ldots, A_{1n}(\lambda_\beta)]^{\mathrm{T}}.$$

It is clear that each $\mathbf{v}_\beta$ is non-zero and $A\mathbf{v}_\beta - \lambda_\beta \mathbf{v}_\beta = A(\lambda_\beta)\mathbf{v}_\beta = \mathbf{0}$. Thus $\mathbf{v}_\beta$ is exactly the unique eigenvector (up to a scale multiplier) of A, related to λ_β.

Now that if $\mathtt{P}_2$ is non-terminating, by Corollary 1, there exists a real eigenvector $\mathbf{v}_\alpha$ of A corresponding to a positive eigenvalue such that $B\mathbf{v}_\alpha \geq 0$. That is to say, if we denote the kth row of B by $\mathbf{b}_k = (b_{k1}, \ldots, b_{kn})$ and set $u_k = \mathbf{b}_k \cdot \mathbf{v}_\alpha$, then $u_k \geq 0$ $(k = 1, \ldots, m)$. We need only to show that each u_k is non-zero. Set

$$\mathbf{v}(\lambda) = [A_{11}(\lambda),\ A_{12}(\lambda), \ldots, A_{1n}(\lambda)]^{\mathrm{T}},$$

and $u_k(\lambda) = B_k \cdot \mathbf{v}(\lambda)$, $(k = 1, \ldots, m)$. Then $u_k = u_k(\lambda_\alpha)$.

Note that, unless $u_k(\lambda)$ is a zero polynomial, $u_k(\lambda)$ does not equal to zero at λ_α because it is a polynomial in λ of degree at most $n-1$ but λ_α is a root of an irreducible polynomial of degree n. We continue to show that $u_k(\lambda)$ is not a zero polynomial either.

If some $u_k(\lambda)$ is a zero polynomial, then for all $\beta = 1, \ldots, n$, $u_k(\lambda_\beta) = 0$, i.e., $\mathbf{b}_k \cdot \mathbf{v}_1 = 0$, $\mathbf{b}_k \cdot \mathbf{v}_2 = 0, \ldots, \mathbf{b}_k \cdot \mathbf{v}_n = 0$. Thus $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are linear dependent. However, these eigenvectors must be linear independent because their eigenvalues are pairwise distinct. This is a contradiction. □

By a similar proof as above, we can obtain the following result.

Corollary 3 *If A and B are both matrices on a field of numbers (e.g., the second extension of the field of rational numbers) and the characteristic polynomial of A is irreducible on this field, then the program $\mathtt{P}_2$ is non-terminating if and only if there exists a real eigenvector $\mathbf{v}$ of A, corresponding to a positive eigenvalue λ, such that $B\mathbf{v} > 0$. In other words, iff*

$$\exists\lambda\,\exists\mathbf{v}.\,\lambda \in \mathbb{R}^+ \wedge \mathbf{v} \in \mathbb{R}^n \wedge A\mathbf{v} = \lambda\,\mathbf{v} \wedge B\mathbf{v} > 0. \tag{16}$$

[10] $A_{ij}(\lambda)$ is $(-1)^{i+j}$ times the determinant of the sub-matrix obtained from $A(\lambda)$ by deleting the ith row and jth column.

6.2. Algorithm

Based on Theorem 4, we design the following algorithm IrrDec to determine the termination of $\mathtt{P}_2$ with the assumption that the characteristic polynomial of A is irreducible.

Procedure IrrDec(B, A)

Step 1: Compute the characteristic polynomial of A and denote it by $D(\lambda)$.

Step 2: Compute the algebraic complement minor of every element in the first (or a fixed) row of $A - \lambda I$ (the characteristic matrix of A), respectively and denote them by A_{1i} $(1 \le i \le n)$.

Step 3: For each row of B, compute $u_j = \sum_{k=1}^{n} b_{jk} A_{1k}$ $(1 \le j \le m)$.

Step 4: Construct a semi-algebraic system[11]

$$S : \{D(\lambda) = 0,\ \ \lambda > 0,\ \ u_1 u_2 > 0,\ u_2 u_3 > 0, \ldots, u_{m-1} u_m > 0\}.$$

Note that it is easy to see that S is another presentation of the body of Formula (16) in terms of semi-algebraic system.

Step 5: By computer algebra tools, determine whether S has real solutions, i.e., whether Formula (16) holds. If yes, $\mathtt{P}_2$ is not terminating. Otherwise, it terminates.

Notice that in this paper we will apply to S the computer algebra tool DISCOVERER [YaX05, Xia07]. For other applications of DISCOVERER to problems in program verification, please refer to [YZX05].

6.3. Examples

We first demonstrate how by the above algorithm to determine the termination of the loop in Example 1.

Example 3 It is easy to compute that

$$\begin{aligned} D(\lambda) &= \lambda^2 - 4\lambda + 1, \\ \mathbf{v} &= (A_{11},\ A_{12})^{\mathrm{T}} = (2 - \lambda,\ 1)^{\mathrm{T}}, \\ B\mathbf{v} &= (u_1,\ u_2) = (2 - \lambda + b,\ \lambda - 2 + b), \end{aligned}$$

and to see that $D(\lambda)$ is irreducible. Then we use DISCOVERER to determine whether the following system has real solutions

$$D(\lambda) = 0, \quad \lambda > 0, \quad (2 - \lambda + b)(\lambda - 2 + b) > 0.$$

The problem can be solved by several symbolic algorithms and functions in DISCOVERER. Say, by applying **nearsolve** for this constant SAS, we get that the number of real solutions of the above system is 0. Thus the loop in Example 1 terminates.

We further illustrate how to use Theorem 4 to decide the termination of linear programs by the following example.

Example 4 Consider the termination of the program **while** $(B\mathbf{x} > 0)$ $\{\mathbf{x} := A\mathbf{x}\}$, where

$$A = \begin{bmatrix} 3 & 1 & 4 & 1 & 5 \\ 9 & 2 & 6 & 5 & 3 \\ 5 & 8 & 9 & 7 & 9 \\ 3 & 2 & 3 & 8 & 4 \\ 6 & 2 & 6 & 4 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 3 & -8 & 3 & 2 & -7 \\ 1 & -4 & 1 & 4 & -2 \\ 4 & -2 & 8 & -5 & 7 \end{bmatrix}.$$

[11] If you like, you can construct two semi-algebraic systems. One has all u_i positive, and the other has all u_i negative.

The characteristic polynomial of A is also irreducible. By Theorem 4, we first compute the algebraic complement minors as follows

$$\begin{aligned}
A_{11}(\lambda) &= -48 + 313\lambda + 8\lambda^2 - 22\lambda^3 + \lambda^4,\\
A_{12}(\lambda) &= 381 + 243\lambda - 117\lambda^2 + 9\lambda^3,\\
A_{13}(\lambda) &= 74 - 539\lambda + 82\lambda^2 + 5\lambda^3,\\
A_{14}(\lambda) &= 144 - 60\lambda + 15\lambda^2 + 3\lambda^3,\\
A_{15}(\lambda) &= -498 + 204\lambda - 54\lambda^2 + 6\lambda^3.
\end{aligned}$$

Construct $\mathbf{v}(\lambda) = (A_{11}(\lambda), \ldots, A_{15}(\lambda))^{\mathrm{T}}$, where λ is an eigenvalue of A. Compute the u_k in Theorem 4 as follows

$$\begin{aligned}
u_1 &= 3A_{11}(\lambda) - 8A_{12}(\lambda) + 3A_{13}(\lambda) + 2A_{14}(\lambda) - 7A_{15}(\lambda) = 804 - 4170\lambda + 1614\lambda^2 - 159\lambda^3 + 3\lambda^4,\\
u_2 &= A_{11}(\lambda) - 4A_{12}(\lambda) + A_{13}(\lambda) + 4A_{14}(\lambda) - 2A_{15}(\lambda) = 74 - 1846\lambda + 726\lambda^2 - 53\lambda^3 + \lambda^4,\\
u_3 &= 4A_{11}(\lambda) - 2A_{12}(\lambda) + 8A_{13}(\lambda) - 5A_{14}(\lambda) + 7A_{15}(\lambda) = -4568 - 1818\lambda + 469\lambda^2 - 39\lambda^3 + 4\lambda^4.
\end{aligned}$$

By Theorem 4, the program is non-terminating if and only if the following semi-algebraic system has real solutions

$$\{D(\lambda) = 0,\ \lambda > 0,\ u_1u_2 > 0,\ u_2u_3 > 0\}, \tag{17}$$

where $D(\lambda)$ is the characteristic polynomial of A.

Using DISCOVERER, say **nearsolve** again, we can conclude that the system (17) has no real solutions. Therefore, the program is terminating. If we delete a constraint and set

$$B = \begin{bmatrix} 3 & -8 & 3 & 2 & -7 \\ 1 & -4 & 1 & 4 & -2 \end{bmatrix},$$

by calling DISCOVERER, we can conclude that the system $\{D(\lambda) = 0,\ \lambda > 0,\ u_1u_2 > 0\}$ has 2 distinct real solutions. That is to say, the resulting program is non-terminating.

6.4. Complexity analysis

In this subsection we will show that the complexity of `IrrDec` is polynomial in n and m, where n is the number of variables of $\mathtt{P}_2$ and m is the number of its Boolean conditions.

In step 1, $D(\lambda)$ can be computed by at most $O(n^3)$ multiplications.

In step 2, all the algebraic complement minors can be computed by at most $O(n^4)$ multiplications.

Step 3 needs at most n^2m multiplications because each u_i is of degree at most $n-1$ and thus has at most n terms.

For steps 4 and 5, we shall analyze the complexity of determining whether the semi-algebraic system

$$S : \{D(\lambda) = 0,\ \lambda > 0,\ u_1u_2 > 0, u_2u_3 > 0, \ldots, u_{m-1}u_m > 0\}$$

has real solutions. Because $D(\lambda)$ (of degree n) is irreducible and each u_i is of degree at most $n-1$, $D(\lambda)$ has no common roots with each u_i.

First, we obtain the isolated intervals, $[a_j, b_j]$, of the positive roots of $D(\lambda)$, where j is at most n. From Sect. 5, we know that the complexity for this step is $O(n^6)$.

Second, as in Sect. 5, we take use of Sturm–Tarski sequences to determine the signs of u_i at those zeros, respectively. For each u_i, we compute the Sturm–Tarski sequence of $D(\lambda)$ w.r.t. u_i that, as shown in Sect. 5, costs $O(n^3)$. So, for all u_is, the cost is $O(mn^3)$. To check the sign changes of those Sturm–Tarski sequences and thus the signs of all u_is at the zeros, we need $O(mn)$ operations for each interval $[a_j, b_j]$ and, obviously, $O(mn^2)$ operations for all the intervals. Therefore, the complexity for this sub-step is $O(mn^3) + O(mn^2) = O(mn^3)$.

Now, we can see that the cost of determining whether S has real solutions dominates the computational complexity of the above procedure, which is $\max(O(n^6), O(mn^3))$. It's clear that the complexity of `IrrDec` is much lower than that of the decision procedure given in [Tiw04] which is $O(n^{m+3})$, and that of `DecTerm` given in this paper which is $\max\{O(n^6), O(n^{m+3})\}$.

7. Conclusions and discussion

This paper present a symbolic decision procedure for the termination of a class of linear programs that was first proved to be decidable by Tiwari in [Tiw04]. In our approach, rounding errors caused by floating-point computation is therefore avoided. We also analyze that the complexity of the new procedure is $\max\{O(n^6), O(n^{m+3})\}$, where n is the number of variables of the program and m is the number of its Boolean conditions. Our algorithm is as efficient as Tiwari's and can be seen as a cute symbolic implementation of Tiwari's decision procedure. In addition, we invent a more efficient symbolic algorithm for the case when the characteristic polynomial of the assignment matrix is irreducible, whose complexity is $\max(O(n^6), O(mn^3))$.

Discussion

1. In fact, it is not hard to invent a symbolic decision procedure for $\mathtt{P}_1$ directly from Tiwari's procedure. This can be done via: Firstly, in his first step, we replace numerically computing eigenvalues and eigenvectors of assignment matrix A with a symbolic computation procedure; Secondly, in his third step, non-deterministically generate a set of non-linear symbolic constraints instead of a set of linear constraints. The big problem with such a symbolic procedure is the high complexity to solve the resulting non-linear symbolic constraints which is at least double exponential in m and n.
2. Any decision procedure that involves floating-point calculation may be not sound at implementation level. This paper, based on [Tiw04], develops a fully symbolic decision procedure for the termination of linear programs, so that we can avoid floating-point computations in termination analysis. Another interesting issue is how to guarantee a (proved) terminating linear program will indeed terminate, when it runs under a compiler with certain precision of floating point computation. Theoretically, if $\mathtt{Q}_2$ in Example 1 does not terminate for an input x, then $\mathtt{Q}_2$ will not terminate for the input Qx either. In Example 1 we use Maple 11 with Digits 10 to calculate A^* and B^* and find $\mathtt{Q}_2$ does not terminate for input $x = (1, -1)$. We should have assumed that Qx can show a run-time non-terminating error in 10 decimal digits of precision for $\mathtt{Q}_1$ that is proved terminating in Example 3. However, when we run it, it surprisingly terminates in the presumed precision. Avoiding run-time errors is very interesting but seems hard to be manipulated. It deserves further and deeper investigation.
3. One may ask whether the assumption in Theorem 4 ("the characteristic polynomial of A is irreducible") can be removed or weakened like "the characteristic polynomial of A is square-free". The following example gives a negative answer.

   Example 5 Given a program as follows

   $$\mathtt{Q}_4 \quad \textbf{while} \quad (B\mathbf{x} > 0) \quad \{\mathbf{x} := A\mathbf{x}\}.$$

   $$\text{where} \quad A = \begin{bmatrix} 4 & 5 & 2 \\ 9 & -1 & -8 \\ 3 & 2 & 3 \end{bmatrix}, B = \begin{bmatrix} 7 & -7 & -6 \\ 1 & 5 & 1 \end{bmatrix},$$

   The characteristic polynomial of A is $\lambda^3 - 6\lambda^2 - 30\lambda + 161 = (\lambda - 7)(\lambda^2 + \lambda - 23)$ which is reducible with eigenvalues $\lambda_1 = 7, \lambda_2 = -\frac{1}{2} + \frac{\sqrt{93}}{2}$ and $\lambda_3 = -\frac{1}{2} - \frac{\sqrt{93}}{2}$. The corresponding eigenvectors are

   $$\mathbf{v}_1 = [8,\ 2,\ 7]^{\mathrm{T}}, \mathbf{v}_2 = [14,\ -17 + \sqrt{93},\ 11 + \sqrt{93}]^{\mathrm{T}}, \mathbf{v}_3 = [14,\ -17 - \sqrt{93},\ 11 - \sqrt{93}]^{\mathrm{T}},$$

   where $\mathbf{v}_1, \mathbf{v}_2$ correspond to positive eigenvalues λ_1 and λ_2, respectively. Please note that the characteristic polynomial of A is square-free and A is non-singular. It is easy to check that $(B\mathbf{v}_1 > 0) \wedge (B\mathbf{v}_2 > 0)$ does not hold and nor $(-B\mathbf{v}_1 > 0) \wedge (-B\mathbf{v}_2 > 0)$. So we may draw a conclusion according to Theorem 4 that the program terminates. Unfortunately the conclusion is wrong, and we can prove the program non-terminating by the algorithm given in Sect. 3.

4. It is quite interesting to investigate whether our approach can be applied to other algorithms that are also suffered from rounding error problems of floating-point computation. We would like to take such a problem as our future work.

Acknowledgments

We are so grateful to Prof. Chaochen Zhou for many fruitful discussions with him on this work and his valuable comments on the draft of this paper. We also thank the anonymous referees for their comments and criticisms which help us to improve the presentation of this paper. This work is supported in part by the grants NSFC-60573007 and NSFC-90718041. The first two authors are also partly supported by the grants NKBRFC-2004CB318003 and NKBRFC-2005CB321902, and the third author is also partly supported by the grants NSFC-60721061, NSFC-60736017, NSFC-60970031 and RCS2008K001.

References

[CLL08] Chen C, Lemaire F, Li L, Moreno Maza M, Pan W, Xie Y (2008) The Constructible Set Tools and Parametric Systems Tools modules of the Regular Chains library in Maple. In: Proceedings of the international conference on computational science and applications, pp 342–352. IEEE Computer Society Press, New York

[CoH91] Collins GE, Hong H (1991) Partial cylindrical algebraic decomposition for quantifier elimination. J Symb Comput 12:299–328

[CoL83] Collins GE, Loos R (1982) Real zeros of polynomials. In: Buchberger B, Collins GE, Loos R (eds) Computer algebra: symbolic and algebraic computation, pp 83–94. Springer, New York

[DoS97] Dolzman A, Sturm T (1997) REDLOG: computer algebra meets computer logic. ACM SIGSAM Bull 31(2):2–9

[HoK71] Hoffman E, Kunze R (1971) Linear algebra, 2nd edn. Prentice-Hall

[Joh98] Johnson JR (1998) Algorithms for polynomial real root isolation. In: Caviness BF, Johnson JR (eds) Quantifier elimination and cylinderical algebraic decomposition, pp 269–299. Springer, Berlin

[Ari96] Lions JL (1996) The ARIANE 5 Flight 501 failure report, 19 July 1996. European Space Agency (ESA)

[MiM82] Miller RK, Michel AN (1982) Ordinary differential equations. Academic Press, New York

[Min05] Mine A (2005) Relational abstract domains for the detection of floating-point run-time eorrors. In: Proceedings of ESOP'05. Lecture notes in computer science, vol 2986, pp 3–17

[Ske92] Skeel R (1992) Roundoff error and the Patriot missile. SIAM News 25(4):11

[Tiw04] Tiwari A (2004) Termination of linear programs. In: Proceedings of CAV'04. Lecture notes in computer science, vol 3114, pp 70–82

[Tod92] Todd M (1992) A low complexity interior-point algorithm for linear programming. SIAM J Optim 2(2):198–209

[XiY02] Xia B, Yang L (2002) An algorithm for isolating the real solutions of semi-algebraic systems. J Symb Comput 34:461–477

[Xia07] Xia B (2007) DISCOVERER: a tool for solving semi-algebraic systems. Software Demo at ISSAC 2007, Waterloo, July 30, 2007. ACM SIGSAM Bull 41(3):102–103

[Yan99] Yang L (1999) Recent advances on determining the number of real roots of parametric polynomials. J Symb Comput 28: 225–242

[YHZ96] Yang L, Hou X, Zeng Z (1996) A complete discrimination system for polynomials. Sci China (Ser E) 39:628–646

[YaX05] Yang L, Xia B (2005) Real solution classifications of a class of parametric semi-algebraic systems. In: Proceedings of international conference on algorithmic algebra and logic, pp 281–289

[YZX05] Yang L, Zhan N, Xia B, Zhou C (2005) Program verification by using DISCOVERER. In: Proceedings of VSTTE'05. Lecture notes in computer science, vol 4174, pp 575–586

Received 30 September 2008
Revised 15 November 2009
Accepted 23 November 2009 by Zhiming Liu and Jim Woodcock
Published online 17 December 2009