Proving Nonnegativity of Polynomial with Computer

Haokun Li, Bican Xia and Lu Yang*
*Corresponding Author: xbc@math.pku.edu.cn
Haokun Li: ker@pku.edu.cn,
School of Mathematical Sciences, Peking University
Beijing, 100871, China
Lu Yang: luyang@casit.ac.cn
CICA, Chinese Academy of Sciences
Chengdu, 610041, China

Abstract

Proving a polynomial is nonnegative (either globally or under some constraints) is a basic and important problem in the field of real algebraic geometry and has many applications in other fields. This article reviews some of our works on this topic with emphasis on how to use our tools to prove nonnegativity of polynomial with computer. Furthermore, a concept of block SOS decomposable polynomials *is introduced which characterizes a class of polynomials whose SDP matrices (corresponding to their SOS decompositions) can be block-diagonalized. It is proved that the set of block SOS decomposable polynomials is measure zero in the set of SOS polynomials.*

1 Introduction

Let us begin with two simple examples.

Example 1 *Prove that* $P \geq 0$ *where*

$$\begin{aligned} P = \ & y^4z^6 + 6x^3y^2z^4 + 9x^6z^2 - 2x^3y^2z^3 - 6x^6z + x^6 + y^2z^4 - 2xy^2z^2 + x^2y^2 \\ & +10xyz^3 - 10x^2yz + 25x^2z^2 + x^2 + 4xy - 6xz + 4y^2 - 12yz + 9z^2. \end{aligned}$$

If P can be written as a sum of squares of polynomials (SOS), it is obviously a certificate for P to be nonnegative. So, we may try SOS decomposition of P first to detect its nonnegativity. By Gram matrix representation [3], the problem of SOS decomposition can be transformed as a problem of semi-definite programming (SDP), which can be solved symbolically or numerically. Actually, there exist some well-known free available SOS tools which are based on numerical SDP solvers [12, 10, 18, 21]. Numerical SOS decomposition algorithms can handle big scale problems and may be used to get exact results [6]. It should be mentioned that, from an algorithmic point of view, writing a multivariate polynomial as an SOS is a crucial part of many applications, see for example [20, 13, 8, 7, 17].

Our tool `SparseSOS` *[22] is free available through https://gitlab.com/haokunli/sparsesos. In order to use* `SparseSOS` *to detect nonnegativity of P, we first express the polynomial P by* $+, -, *, \hat{}$ *, integers and variables in a file, say demo.txt, as follows:*

```
y^4*z^6+6*x^3*y^2*z^4+9*x^6*z^2-2*x^3*y^2*z^3-6*x^6*z+x^6+y^2*z^4
-2*x*y^2*z^2+10*x*y*z^3+x^2*y^2-10*x^2*y*z+25*x^2*z^2+x^2+4*x*y
-6*x*z+4*y^2-12*y*z+9*z^2
```

Then, we only need to type in:

```
is_sos demo.txt
```

to run `SparseSOS`*. The output is "successful" which means the polynomial P can be an SOS.*

Of course, nonnegative polynomials are not necessarily SOS. Actually, the number of nonnegative polynomials is much larger than that of SOS polynomials [1]. Nevertheless, trying SOS decomposition is an efficient way to detect nonnegativity. In Section 2, we will review our recent work in [22] and report some random examples solved by the new sparse SOS decomposition algorithm `SparseSOS`.

Sparse SOS decomposition is more efficient in practice because the problem can be reduced to smaller ones in some cases, *i.e.* the SDP matrix corresponding to the problem can be block-diagonalized. In Section 3, we define a concept of *block SOS decomposable polynomials* which is a generalization of those special classes in [5] and [14]. Roughly speaking, it is a class of polynomials whose SOS decomposition problem can be transformed into smaller ones by considering their supports only (coefficients are not considered). We prove that the set of block SOS decomposable polynomials is measure zero in the set of SOS polynomials.

Example 2 *[28] Prove that $f \geq 0$ under the constraints that $a \geq 0, b \geq 0, c \geq 0, abc - 1 = 0$, where*

$$\begin{aligned} f = \; & 2b^4c^4 + 2b^3c^4a + 2b^4c^3a + 2b^3c^3a^2 + 2a^3c^3b^2 + 2a^4c^3b + 2a^3c^4b + 2a^4c^4 \\ & +2a^3b^4c + 2a^4b^4 + 2a^3b^3c^2 + 2a^4b^3c - 3b^5c^4a^3 - 6b^4c^4a^4 - 3b^5c^3a^4 \\ & -3b^4c^3a^5 - 3b^4c^5a^3 - 3b^3c^5a^4 - 3b^3c^4a^5. \end{aligned}$$

Generally, we know that

$$\phi \Rightarrow \psi \iff \neg\phi \vee \psi \iff \neg(\phi \wedge \neg\psi).$$

So, proving $\phi \Rightarrow \psi$ is equivalent to proving $\phi \wedge \neg\psi$ is inconsistent. Therefore

$$a \geq 0 \wedge b \geq 0 \wedge c \geq 0 \wedge abc - 1 = 0 \Longrightarrow f \geq 0$$

is equivalent to the following system is inconsistent

$$a \geq 0 \wedge b \geq 0 \wedge c \geq 0 \wedge abc - 1 = 0 \wedge f < 0.$$

This is a typical quantifier elimination (QE) problem [19, 4] and may be solved by some general QE tools, such as `QEPCAD` and `REDLOG`.

Yang *et. al.* [26] gave a theorem for explicitly determining the condition for a given polynomial to have a given number of real (and/or complex) zeros. Sometimes the conditions are called the root-classification of polynomials. With this theorem and its generalization to the case of semi-algebraic system, Yang *et. al.* proposed an algorithm for proving and discovering inequality-type theorems automatically [25, 27, 24]. Indeed, the algorithm solves a special kind of QE problems which have at least one polynomial equation. This algorithm has been improved and implemented as a Maple package DISCOVERER [23] by Xia. Since 2009, the main functions of DISCOVERER have been integrated into the `RegularChains` library of Maple. Since then, the implementation has been improved by Chen *et. al.* [2].

To prove the example by Maple, we first start Maple and load several packages as follows.

```
> with(RegularChains):
> with(ParametricSystemTools):
> with(SemiAlgebraicSetTools):
```

Then define an order of the unknowns:

```
> R := PolynomialRing([a, b, c]);
```

To get more information from the output of the function directly, we type in:

```
> infolevel[RegularChains]:=1;
```

Now, by calling

```
> RealRootClassification([abc-1], [a, b, c], [-f], [ ], 2, 0, R);
```

we will know at once that the inequality holds.

We will give in detail an introduction on how to use the function `RealRootClassification` (`RRC` for short) to prove a polynomial is nonnegative under some polynomial inequality and/or equation constraints in Section 4.

2 Sparse SOS Decomposition via `SparseSOS`

For polynomial $f \in \mathbb{R}[\mathbf{x}] = \mathbb{R}[x_1, \ldots, x_n]$, the support of f, denoted by $S(f)$, is

$$\{\alpha \in \mathbb{N}^n \mid \text{the coefficient of } \mathbf{x}^\alpha \text{ is not 0 in } f\}.$$

f is SOS if and only if there exist polynomials $q_1, \ldots, q_s \in \mathbb{R}[\mathbf{x}]$ satisfying $f(\mathbf{x}) = \sum_{i=1}^{s} q_i^2$.

For a polynomial $f \in \mathbb{R}[\mathbf{x}]$ and a given monomial basis $M = \{\mathbf{x}^{\alpha_1}, \ldots, \mathbf{x}^{\alpha_r}\}$, the SOS condition for f can be converted to the problem of deciding if there exists a positive semidefinite matrix Q (Gram matrix) such that $f = M^T Q M$ which can be effectively solved by semidefinite programming (SDP) [13].

When the given polynomial has many variables and a high degree, the corresponding SDP problem is hard to be dealt with by existing SDP solvers due to the very large size of the corresponding SDP matrix. On the other hand, most polynomials coming from practice have certain structures including symmetry and sparsity. So it is very important to take full advantage of structures of polynomials to reduce the size of corresponding SDP problems. In recent years, a lot of work has been done on this subject, see foe example [16, 11, 7, 5, 14].

We proposed a new sparse SOS decomposition algorithm `SparseSOS` in [22] which is based on a new sparsity pattern, called *cross sparsity patterns*. Given a polynomial $f \in \mathbb{R}[\mathbf{x}]$ with the support $\mathrm{A} \subseteq \mathbb{N}^n$ and a monomial basis $\mathrm{B} = \{\mathbf{x}^{\alpha_1}, \ldots, \mathbf{x}^{\alpha_r}\}$, the cross sparsity pattern associated with A is represented by an $r \times r$ symmetric $(0,1)$-matrix R_{A} whose elements are defined by

$$R_{ij} = \begin{cases} 1, & \alpha_i + \alpha_j \in \mathrm{A} \cup 2\mathrm{B}, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where $2\mathrm{B} = \{2\alpha_1, \dots, 2\alpha_r\}$.

Following the chordal sparsity approaches, we associate the matrix R_{A} with an undirected graph $G(V_{\mathrm{A}}, E_{\mathrm{A}})$ where

$$V_{\mathrm{A}} = \{1, 2, \dots, r\} \text{ and } E_{\mathrm{A}} = \{\{i, j\} \mid i, j \in V_{\mathrm{A}}, i < j, R_{ij} = 1\}$$

and generate a chordal extension of $G(V_{\mathrm{A}}, E_{\mathrm{A}})$. Then as usual, we use matrix decompositions for positive semidefinite matrices with chordal sparsity patterns to construct sets of supports for a blocking SOS decomposition. We prove that the blocking SOS decomposition obtained by cross sparsity patterns is always a refinement of the block-diagonalization obtained by the sign-symmetry method [11].

The algorithm has been implemented with C++ as a tool also named `SparseSOS` which uses mosek 8.1 as LP and SDP solver. It turns out that the new algorithm dramatically reduces the computational cost compared to existing tools and can handle some really huge polynomials which are unsolvable by any existing SOS solvers even exploiting sparsity. In the following, we only report experiments on some randomly generated polynomials. For more examples and comparison to other tools, see [22].

A randomly generated sparse SOS polynomial

$$f = \sum_{i=1}^{k} f_i^2 \in \mathbf{randpoly}(n, d, k, p)$$

is constructed as follows:

- select monomials from $\mathbf{x}^{\mathbb{N}_d^n}$ with probability p to generate a monomial set M;
- randomly assign the elements of M to $f_1, \dots, f_k$ with coefficients between -10 and 10.

We generate 12 random polynomials $p_1, \dots, p_{12}$, where

$$\begin{aligned} p_1, p_2, p_3 &\in \mathbf{randpoly}(10, 6, 10, 0.005), \\ p_4, p_5, p_6 &\in \mathbf{randpoly}(10, 6, 10, 0.01), \\ p_7, p_8, p_9 &\in \mathbf{randpoly}(10, 6, 10, 0.015), \\ p_{10}, p_{11}, p_{12} &\in \mathbf{randpoly}(10, 8, 20, 0.002). \end{aligned}$$

For example,

$$\begin{aligned} p_2 = \; & 16x_0^2x_1^8x_5^2 + 49x_0^2x_3^2x_4^4x_5^4 + 98x_0x_1^2x_2x_3^2x_4^3x_5^2x_6 + 49x_1^4x_2^2x_3^2x_4^2x_6^2 + 81x_1^4x_2^4x_5^2x_6^2 \\ & +25x_4^6x_5^2x_6^4 - 20x_0x_1x_3^2x_4^4x_5x_6^2x_7 + 98x_0x_2x_3^2x_4^2x_5^3x_6^2x_7 + 98x_1^2x_2^2x_3^2x_4x_5x_6^3x_7 \\ & +4x_0^2x_1^2x_3^4x_4^2x_7^2 + x_1^6x_2^2x_5^2x_7^2 + 49x_2^2x_3^2x_5^2x_6^4x_7^2 + 36x_2^2x_4^4x_5^2x_7^4 + 4x_4^4x_5^2x_7^6 + 81x_4^6x_6^2x_7^2x_8^2 \\ & +18x_0x_1^3x_2^3x_5^3x_6x_8 - 8x_0^2x_1^4x_2x_4^2x_5x_7x_8 - 56x_0x_1^4x_2^2x_4x_5^2x_7x_8 + 18x_1^3x_2x_4^3x_5x_6x_7^2x_8 \\ & +x_0^2x_1^2x_2^2x_5^4x_8^2 + 16x_0^2x_1^2x_2^2x_5^2x_6^2x_8^2 + x_0^2x_2^2x_4^4x_7^2x_8^2 + 14x_0x_2^3x_4^3x_5x_7^2x_8^2 + 49x_2^4x_4^2x_5^2x_7^2x_8^2 \\ & +10x_2x_3x_4^3x_5x_6^2x_7^2x_8^2 - 4x_0x_1x_2x_3^3x_4x_7^3x_8^2 - 80x_0^2x_1x_2x_5^2x_6x_7^3x_8^2 + 100x_0^2x_5^2x_7^6x_8^2 \\ & -96x_2x_3^2x_4^2x_5^2x_7^2x_8^3 + x_2^2x_3^2x_7^4x_8^4 + 64x_3^4x_5^2x_8^6 + 72x_0^4x_1x_2x_5x_6^3x_8x_9 - 180x_0^4x_5x_6^2x_7^3x_8x_9 \\ & +x_2^2x_7^6x_8^2x_9^2 + 81x_3^2x_4^4x_8^4x_9^2 + 100x_0^4x_6^2x_8^4x_9^2 - 12x_2x_3^2x_4x_7^2x_8^4x_9^2 + 36x_3^2x_4^2x_8^4x_9^4 \end{aligned}$$

$$
\begin{aligned}
&+12x_2^2x_4^2x_5x_7^5x_8x_9 + 80x_0^3x_1^4x_5x_6x_8^2x_9 - 20x_0^3x_2x_4^2x_6x_7x_8^3x_9 - 140x_0^2x_2^2x_4x_5x_6x_7x_8^3x_9 \\
&-16x_2x_3^2x_5x_7^3x_8^4x_9 + 4x_0^2x_1^4x_2^2x_6^2x_9^2 + 81x_0^6x_6^4x_9^2 - 60x_3x_4^4x_5x_6^2x_8^2x_9^2 + 24x_0x_1x_3^3x_4^2x_7x_8^2x_9^2 \\
&+80x_0^2x_2^2x_3x_4^3x_5x_6^2 + 42x_0^2x_3^2x_4^3x_5^2x_6^2 + 42x_0x_1^2x_2x_3^2x_4^2x_6^3 - 32x_0^3x_1x_2^2x_3^3x_4x_7 - 18x_0^2x_1^3x_2^2x_5^3x_7 \\
&+126x_0x_1^2x_3^2x_4^2x_5^3x_7 + 126x_1^4x_2x_3^2x_4x_5x_6x_7 + 42x_0x_2x_3^2x_4x_5x_6^4x_7 + 126x_1^2x_2x_3^2x_5^2x_6^2x_7^2 \\
&-28x_1x_4^2x_5^2x_6^3x_7^3 - 64x_0^2x_1x_2x_3x_4^2x_5^2x_6x_8 - 162x_0^2x_2x_4^3x_5^2x_6x_7x_8 + 90x_1x_3^2x_4^3x_5x_6^2x_7x_8 \\
&-36x_0x_1^2x_3^4x_4x_7^2x_8 + 160x_0^2x_3x_4^2x_5^2x_7^3x_8 + 16x_0^2x_2^3x_3^2x_7^2x_8^2 - 80x_0x_1x_2x_5x_6^4x_8^3 \\
&+72x_2x_3^2x_4^2x_5x_7^2x_8^3 + 18x_1x_2x_3^3x_7^3x_8^3 + 200x_0x_5x_6^3x_7^3x_8^3 + 90x_4^3x_5x_6^2x_8^5 - 36x_0x_1x_3^2x_4x_7x_8^5 \\
&-96x_3^4x_5x_8^6 + 18x_2x_3x_7^2x_8^7 - 144x_0^4x_3x_4^2x_5x_6^2x_9 - 40x_0x_1^3x_2^2x_6^4x_9 - 180x_0^3x_6^5x_8^2x_9 + 16x_1^2x_3^4x_4^2 \\
&+180x_2x_3x_4^5x_7x_8^2x_9 + 12x_2x_3^2x_7^3x_8^4x_9 - 8x_1^3x_2x_3x_4x_5x_7x_8x_9^2 - 96x_0^2x_2^2x_3^2x_4x_8^2x_9^2 \\
&-72x_3x_4^4x_6x_7x_8^2x_9^2 - 108x_1x_3^3x_4x_7x_8^3x_9^2 + 36x_2x_3x_4^2x_5x_7x_8^3x_9^2 - 108x_3x_4x_8^7x_9^2 + 64x_0^4x_2^4x_3^2 \\
&+56x_0x_1x_3^3x_4^3x_5^2 + 64x_0^2x_3^2x_4^4x_5^2 + 81x_0^4x_2^2x_5^4 + 56x_1^3x_2x_3^3x_4^2x_6 + 9x_0^2x_3^2x_4^2x_6^4 + 48x_1x_2x_3^2x_4x_8^3 \\
&+100x_1^2x_2^2x_6^6 + 49x_1^2x_5^2x_6^6 + 54x_0x_1^2x_3^2x_4x_5x_6^2x_7 + 56x_1x_2x_3^3x_4x_5x_6^2x_7 + 100x_2^2x_4^6x_7^2 \\
&+81x_1^4x_3^2x_5^2x_7^2 + 144x_0^2x_1x_2^2x_3^3x_7x_8 + 160x_0x_3x_4^2x_5x_6^3x_8^2 + 81x_1^2x_3^4x_7^2x_8^2 + 84x_0x_2x_3x_4^2x_5^2x_8^3 \\
&+84x_1^2x_2^2x_3x_4x_6x_8^3 + 84x_2^2x_3x_5x_6^2x_7x_8^3 + 100x_6^6x_8^4 + 144x_0^2x_2^2x_3x_8^5 + 36x_3^4x_8^6 + 36x_2^2x_8^6 \\
&+162x_1x_3^2x_7x_8^6 + 81x_8^10 + 40x_2^2x_4^3x_5x_7^2x_8x_9 + 72x_0^2x_2x_3x_4x_5^2x_8x_9^2 + 4x_2^2x_5^2x_7^2x_8^2x_9^2 \\
&+16x_3^2x_4^2x_8^2x_9^4 + 24x_0x_1x_3^3x_4^2x_6^2 + 72x_1^3x_3^3x_4x_5x_7 + 36x_0x_2x_3x_4x_6^2x_8^3 + 108x_1^2x_2x_3x_5x_7x_8^3.
\end{aligned}
$$

The 12 polynomials can be found at https://gitlab.com/haokunli/sparsesos/example/ and were proved to be SOS via `SparseSOS` on a computer (6-Core Intel Core i7-8750H@2.20GHz CPU, 32GB RAM memory, ARCH LINUX SYSTEM).

Timings (in second) of `SparseSOS` on the polynomials p_1 to p_{12}.

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}
0.88	0.95	0.98	5.50	5.55	23.91	28.82	1634.52	79.21	9.06	10.27	12.29

3 Block SOS Decomposition

The key idea of taking advantage of sparsity in SOS decomposition is to find those SOS polynomials such that the corresponding SDP matrices can be block-diagonalized. A natural question is that to what extent sparse SOS algorithms work. We define those polynomials as *block SOS decomposable polynomials* and prove that the set of block SOS decomposable polynomials is measure zero in the set of SOS polynomials.

3.1 Notations

For a set $M \subseteq \mathbb{N}^n$, let $\mathrm{conv}(M)$ denote the convex hull of M. For a set $Y \subseteq \mathbb{R}[\mathbf{x}]$, let $Y^2 := \{f^2 \mid f \in Y\}$ and $\sum Y^2 := \{\sum_{i=1}^{s} f_i^2 \mid f_i \in Y \text{ for } i = 1, \dots, s \text{ and } s \in \mathbb{N}\}$. So the set of all polynomials that are SOS can be denoted by $\sum \mathbb{R}[\mathbf{x}]^2$.

For vector spaces A, B, let $A \oplus B := \{(a, b) | a \in A \wedge b \in B\}$, $\bigoplus_{i=1}^{m} A_i := A_1 \oplus \cdots \oplus A_m$. Let $\mathbb{S}^{n\times n}$ denote the set of $n \times n$ symmetric matrices and $\mathbb{S}_+^{n\times n}$ denote the set of $n \times n$ positive semi-definite matrices. $\{0, 1\}^{n\times n}$ denotes the $n \times n$ matrices with $0, 1$ as elements. For vectors a and b, let $\langle a, b\rangle$ denote the inner product of a and b. For two $n \times n$ matrices A, B, let

$\langle A, B\rangle := \sum_{i,j} A_{ij}B_{ij} = \mathrm{Tr}(A^{\mathrm{T}}B)$. For a given $k \in \mathbb{N}$ and a given $Q \subseteq \mathbb{N}^n$, let

$$\begin{aligned}\Lambda(k) &:= \{(a_1, ..., a_n) \in \mathbb{N}^n \mid \sum_{i=1}^{n} a_i \leq k\},\\ \mathbb{R}[\mathbf{x}]_k &:= \{f \in \mathbb{R}[\mathbf{x}] \mid \deg(f) \leq k\},\\ \mathbb{R}[\mathbf{x}]_Q &:= \{f \in \mathbb{R}[\mathbf{x}] \mid S(f) \subseteq Q\},\\ \sigma(Q) &:= \{a \in Q \mid \forall b, c \in Q, b + c \neq 2a \vee b = c = a\}.\end{aligned}$$

So $\sum \mathbb{R}[\mathbf{x}]_Q^2$ is the set of polynomials which can be represented as sum of squares of the polynomials in $\mathbb{R}[\mathbf{x}]_Q$.

Definition 3 ([5]) *Suppose $f \in \mathbb{R}[\mathbf{x}]$ and $Q \subseteq \mathbb{N}^n$. If*

$$f \textit{ is SOS} \Rightarrow \exists \{f_i\}_{i=1}^{s} (f = \sum_{i=1}^{s} f_i^2 \wedge S(f_i) \subseteq Q),$$

then Q is said to be an SOS support *of f and denoted by* $\mathrm{SOSS}(f, Q)$.

Definition 4 *Given $T \subseteq \mathbb{N}^n$ and $Q_i \subseteq \mathbb{N}^n$ for $i = 1, \ldots, m$. The set $\{Q_i\}_{i=1}^m$ is said to be a* block SOS support *of T if*

1. $\{f \mid f \in \sum \mathbb{R}[\mathbf{x}]^2, S(f) = T\} \subseteq \sum_{i=1}^m \sum \mathbb{R}[\mathbf{x}]_{Q_i}^2$ *and*
2. $Q_i \cap Q_j = \emptyset$ for $i \neq j$.

A block SOS support of $S(f)$ where $f \in \mathbb{R}[\mathbf{x}]$ is also called a block SOS support *of f. Suppose $\{Q_i\}_{i=1}^m$ is a block SOS support of $T \subseteq \mathbb{N}^n$. If for any $Q_i \neq \emptyset$ such that*

$$\exists f \in \mathbb{R}[\mathbf{x}](S(f) = T \wedge \neg \mathrm{SOSS}(f, Q_i)),$$

then $\{Q_i\}_{i=1}^m$ is said to be a proper block SOS support *of T and T is* block SOS decomposable. *Similarly, for $f \in \mathbb{R}[\mathbf{x}]$, if $\{Q_i\}_{i=1}^m$ is a proper block SOS support of $S(f)$, then $\{Q_i\}_{i=1}^m$ is said to be a* proper block SOS support *of f and f is* block SOS decomposable.

For some relations between block SOS decomposition and the work of [5, 14], see [9].

3.2 $\sum_{i=1}^m \sum \mathbb{R}[\mathbf{x}]_{Q_i}^2$ is Closed

For arbitrarily given $k \in \mathbb{N}$ and a finite set $Q \subset \mathbb{N}^n$, when we talk about the topology of $\mathbb{R}[\mathbf{x}]_k$ in this paper, we mean the topology of finite-dimensional vector space $\mathbb{R}^{\#\Lambda(k)}$. Similarly, the topology of $\mathbb{R}[\mathbf{x}]_Q$ is the topology of finite-dimensional vector space $\mathbb{R}^{\#Q}$.

For a given $k \in \mathbb{N}$, we prove in this section that the polynomial space $\sum_{i=1}^m \sum \mathbb{R}[\mathbf{x}]_{Q_i}^2$ is closed in $\mathbb{R}[\mathbf{x}]_{2k}$, where $Q_i \subseteq \Lambda(k)$.

Lemma 5 *If $A \in \mathbb{S}_+^{m\times m}$, then there exist $v_i \in \mathbb{R}^{m\times 1}$ $(i = 1, \ldots, m)$ such that $A = \sum_{i=1}^m v_i v_i^{\mathrm{T}}$.*

Proof. By using Cholesky decomposition, we have $A = LL^{\mathrm{T}} = \sum_{i=1}^m L_i L_i^{\mathrm{T}}$ where $L \in \mathbb{R}^{m\times m}$ and L_i is the i^{th} column of L. Let $v_i := L_i$. ■

Lemma 6 *Suppose $Q \subset \mathbb{N}^n$ and $m = \#Q < +\infty$. Then*

$$\sum \mathbb{R}[\mathbf{x}]_Q^2 = \{\langle (\mathbf{x}^{a+b})_{(a,b)\in Q\times Q}, G\rangle \mid G \in \mathbb{S}_+^{m\times m}\}.$$

Proof. For $G \in \mathbb{S}_+^{m\times m}$, by Lemma 5, there exist $\{v_i\}_{i=1}^m$ such that $G = \sum_{i=1}^m v_i v_i^{\mathrm{T}}$. So $\langle (\mathbf{x}^{a+b})_{(a,b)\in Q\times Q}, G\rangle = \sum_{i=1}^m \langle (\mathbf{x}^{a+b})_{(a,b)\in Q\times Q}, v_i v_i^{\mathrm{T}}\rangle = \sum_{i=1}^m \langle (\mathbf{x}^a)_{a\in Q}, v_i\rangle^2 \in \sum \mathbb{R}[\mathbf{x}]_Q^2$.

On the other hand, for any $\sum_{i=1}^k f_i^2 \in \sum \mathbb{R}[\mathbf{x}]_Q^2$, there exist $v_i \in \mathbb{R}^{m\times 1}$ for each $i = 1, \ldots, k$ such that $f_i = \langle (\mathbf{x}^a)_{a\in Q}, v_i\rangle$. Then

$$\begin{aligned}\sum_{i=1}^k f_i^2 =& \sum_{i=1}^k \langle (\mathbf{x}^a)_{a\in Q}, v_i\rangle^2 = \sum_{i=1}^k \langle (\mathbf{x}^{a+b})_{(a,b)\in Q\times Q}, v_i v_i^{\mathrm{T}}\rangle \\ =& \langle (\mathbf{x}^{a+b})_{(a,b)\in Q\times Q}, \sum_{i=1}^k v_i v_i^{\mathrm{T}}\rangle = \langle (\mathbf{x}^{a+b})_{(a,b)\in Q\times Q}, LL^{\mathrm{T}}\rangle \\ \in& \{\langle (\mathbf{x}^{a+b})_{(a,b)\in Q\times Q}, G\rangle \mid G \in \mathbb{S}_+^{m\times m}\},\end{aligned}$$

where $L = (v_1, \ldots, v_k) \in \mathbb{R}^{m\times k}$. ■

Lemma 7 *Suppose $Q \subset \mathbb{N}^n$ and $m = \#Q < +\infty$. Then*

$$\sum \mathbb{R}[\mathbf{x}]_Q^2 = \{f_1^2 + \cdots + f_m^2 | f_1, ..., f_m \in \mathbb{R}[\mathbf{x}]_Q\}.$$

Proof. By Lemma 6, we know $\sum \mathbb{R}[\mathbf{x}]_Q^2 = \{\langle (\mathbf{x}^{a+b})_{(a,b)\in Q\times Q}, G\rangle \mid G \in \mathbb{S}_+^{m\times m}\}$. By Lemma 5, there exist $\{v_i\}_{i=1}^m$ such that $G = \sum_{i=1}^m v_i v_i^{\mathrm{T}}$. Let $f_i := \langle (\mathbf{x}^a)_{a\in Q}, v_i\rangle$ for $i = 1, \ldots, m$, we have $f = \sum_{i=1}^m f_i^2$. ■

The proof of the following theorem uses the idea of [17].

Theorem 8 *For a given $k \in \mathbb{N}$ and any finite set $\{Q_i\}_{i=1}^m$ where $Q_i \subseteq \Lambda(k)$ for $i = 1, \ldots, m$, $\sum_{i=1}^m \sum \mathbb{R}[\mathbf{x}]_{Q_i}^2$ is closed in $\mathbb{R}[\mathbf{x}]_{2k}$.*

Proof. Let φ be a continuous mapping:

$$\begin{aligned}\varphi : \bigoplus_{\#Q_1} \mathbb{R}[\mathbf{x}]_{Q_1} \times \cdots \times \bigoplus_{\#Q_m} \mathbb{R}[\mathbf{x}]_{Q_m} \to \mathbb{R}[\mathbf{x}]_{2k} \\ ((f_{1,j})_{j\in Q_1}, \ldots, (f_{m,j})_{j\in Q_m}) \mapsto \sum_{i=1}^m \sum_{j\in Q_i} f_{i,j}^2.\end{aligned}$$

By Lemma 7, we know $\sum_{i=1}^m \sum \mathbb{R}[\mathbf{x}]_{Q_i}^2 = \mathrm{rng}(\varphi)$ and $\ker(\varphi) = 0$.

Let V be the image of unit sphere of $\bigoplus_{\#Q_1} \mathbb{R}[\mathbf{x}]_{Q_1} \times \cdots \times \bigoplus_{\#Q_m} \mathbb{R}[\mathbf{x}]_{Q_m}$. Obviously V is compact and $0 \notin V$. Since $\varphi(\lambda z) = \lambda^2 \varphi(z)$ for $\lambda \in \mathbb{R}$, we have $\sum_{i=1}^m \sum \mathbb{R}[\mathbf{x}]_{Q_i}^2 = \sum_{\lambda\in[0,\infty)} \lambda V$.

Given an infinite sequence $\{\lambda_i v_i\}_{i=1,2,\ldots}$ where $\lambda_i \in [0, \infty)$ and $v_i \in V$, assume $\lim_{i\to\infty} \lambda_i v_i = p$. Because V is compact, there exists a convergent subsequence $\{v_{i_j}\} \subseteq \{v_i\}$ such that $v = \lim_{j\to\infty} v_{i_j} \in V$. So $v \neq 0$.

Then $\lim_{j\to\infty} \lambda_{i_j} = \lim_{j\to\infty} \frac{|\lambda_{i_j} v_{i_j}|}{|v_{i_j}|} = \frac{|p|}{|v|} \geq 0$, where $|\cdot|$ stands for $\|\cdot\|_2$. Finally,

$$p = \lim_{j\to\infty} \lambda_{i_j} v_{i_j} = (\lim_{j\to\infty} \lambda_{i_j})(\lim_{j\to\infty} v_{i_j}) = \frac{|p|}{|v|} v \in \sum_{\lambda\in[0,\infty)} \lambda V = \sum_{i=1}^m \sum \mathbb{R}[\mathbf{x}]_{Q_i}^2.$$

So $\sum_{i=1}^m \sum \mathbb{R}[\mathbf{x}]_{Q_i}^2$ is closed. ■

3.3 Sparsity Is Necessary

Firstly, we show that SOS decomposition of a certain class of polynomials is unique by the two lemmas below.

Lemma 9 *For any finite set* $Q \subset \mathbb{N}^n$, $\mathrm{conv}(Q) = \mathrm{conv}(\sigma(Q))$.

Proof. Without loss of generality, suppose $Q \neq \sigma(Q)$. Let $Q \backslash \sigma(Q) = \{a_1, a_2, \ldots, a_r\}$ and

$$Q_0 = Q,\ \ Q_i = Q_{i-1} \backslash \{a_i\} \text{ for } i = 1, \ldots, r.$$

So $Q_r = \sigma(Q)$. We then prove the claim $\mathrm{conv}(Q_i) = \mathrm{conv}(Q)$ for $i = 0, ..., r$ by induction on i.

When $i = 0$, it is obviously true. Suppose the claim holds for $i-1$. It is clear that we need only to prove $a_i \in \mathrm{conv}(Q_i)$.

Since $a_i \in Q \backslash \sigma(Q)$, there exist $b, c \in Q$ such that $b \neq a_i,\ c \neq a_i,\ b + c = 2a_i$. Since $b, c \in Q \subseteq \mathrm{conv}(Q) = \mathrm{conv}(Q_{i-1})$, there exist $\lambda_{b,\alpha}, \lambda_{c,\alpha} \in [0, 1]$ for each $\alpha \in Q_{i-1}$ such that

$$\begin{array}{l} b = \sum_{\alpha \in Q_{i-1}} \lambda_{b,\alpha}\alpha, \ \ \sum_{\alpha \in Q_{i-1}} \lambda_{b,\alpha} = 1, \\ c = \sum_{\alpha \in Q_{i-1}} \lambda_{c,\alpha}\alpha, \ \ \sum_{\alpha \in Q_{i-1}} \lambda_{c,\alpha} = 1. \end{array}$$

Because $b \neq a_i$ and $c \neq a_i$, we have that $\lambda_{b,a_i} + \lambda_{c,a_i} < 2$.

On the other hand, $2a_i = \sum_{\alpha \in Q_{i-1}} \lambda_{b,\alpha}\alpha + \sum_{\alpha \in Q_{i-1}} \lambda_{c,\alpha}\alpha$ because $b + c = 2a_i$. Therefore

$$a_i = \sum_{\alpha \in Q_{i-1} \backslash \{a_i\}} \frac{\lambda_{b,\alpha} + \lambda_{c,\alpha}}{2 - \lambda_{b,a_i} - \lambda_{c,a_i}}\alpha = \sum_{\alpha \in Q_i} \frac{\lambda_{b,\alpha} + \lambda_{c,\alpha}}{2 - \lambda_{b,a_i} - \lambda_{c,a_i}}\alpha.$$

It is easy to see that

$$\frac{\lambda_{b,\alpha} + \lambda_{c,\alpha}}{2 - \lambda_{b,a_i} - \lambda_{c,a_i}} \geq 0 \text{ and } \sum_{\alpha \in Q_i} \frac{\lambda_{b,\alpha} + \lambda_{c,\alpha}}{2 - \lambda_{b,a_i} - \lambda_{c,a_i}} = 1.$$

This means $a_i \in \mathrm{conv}(Q_i)$ that completes the proof. ∎

Lemma 10 *Let* $a, b \in \mathbb{N}^n$ *and* $a \neq b$. *If* $a - b \in \{-1, 0, 1\}^n$, *then any SOS decomposition of* $\mathbf{x}^{2a} + 2\mathbf{x}^{a+b} + \mathbf{x}^{2b}$ *can only be of the form* $\sum_{i=1}^m (\lambda_{i1}\mathbf{x}^a + \lambda_{i2}\mathbf{x}^b)^2$ *where there exists* i *such that* $\lambda_{i1}\lambda_{i2} \neq 0$.

Proof. Suppose $\{f_i\}_{i=1}^m$ satisfy $(\mathbf{x}^a + \mathbf{x}^b)^2 = \sum_{i=1}^m f_i^2$. Let $Q = \cup_{i=1}^m S(f_i)$. By the definition of $\sigma(Q)$, it is not hard to see that the squares of items in $\sigma(Q)$ must appear in $\sum_{i=1}^m f_i^2$ (since they do not eliminate). Therefore $\sigma(Q) \subseteq \{a, b\}$. By Lemma 9, we know

$$Q \subseteq \mathrm{conv}(Q) = \mathrm{conv}(\sigma(Q)) \subseteq \mathrm{conv}(\{a, b\}) = \{a, b\}.$$

The last equality holds because $a - b \in \{-1, 0, 1\}^n$. So $f_i = \lambda_{i1}\mathbf{x}^a + \lambda_{i2}\mathbf{x}^b$. Because $\mathbf{x}^{a+b}$ appears in $\sum_{i=1}^m f_i^2$, there exists i such that $\lambda_{i1}\lambda_{i2} \neq 0$. ∎

Lemma 11 *Let* $k \in \mathbb{N}$, $S_1 \subset S_2 \subseteq \Lambda(2k)$. *If* $\{f \in \sum \mathbb{R}[\mathbf{x}]_k^2 \mid S(f) = S_2\} \neq \emptyset$ *and* $\{Q_i\}_{i=1}^m$ *is a block SOS support of* S_2, *then* $\{Q_i\}_{i=1}^m$ *is also a block SOS support of* S_1.

Proof. Take $f_2 \in \{f \in \sum \mathbb{R}[\mathbf{x}]_k^2 \mid S(f) = S_2\}$. For any $f_1 \in \{f \in \sum \mathbb{R}[\mathbf{x}]_k^2 \mid S(f) = S_1\}$, without loss of generality, suppose the absolute values of the coefficients of f_2 are less than the absolute values of nonzero coefficients of f_1. Let $g_j = \frac{f_2}{j} + f_1$ for any positive integer j. Then $S(g_j) = S_2$ and $\lim_{j\to\infty} g_j = f_1$.

Because $\{Q_i\}_{i=1}^m$ is a block SOS support of S_2, then $\{f \in \sum \mathbb{R}[\mathbf{x}]_k^2 \mid S(f) = S_2\} \subseteq \sum_{i=1}^m \sum \mathbb{R}[\mathbf{x}]_{Q_i}^2$. By Theorem 8, $f_1 \in \sum_{i=1}^m \sum \mathbb{R}[\mathbf{x}]_{Q_i}^2$. So $\{f \in \sum \mathbb{R}[\mathbf{x}]_k^2 \mid S(f) = S_1\} \subseteq \sum_{i=1}^m \sum \mathbb{R}[\mathbf{x}]_{Q_i}^2$. Therefore, $\{Q_i\}_{i=1}^m$ is a block SOS support of S_1. ■

Based on the above two lemmas, we then prove that, for an arbitrarily given degree bound $2k$, the *full polynomials*, *i.e.* polynomials having all monomials with degree no more than $2k$, are not block SOS decomposable.

Theorem 12 *For any $k \in \mathbb{N}$, $\Lambda(2k)$ is not block SOS decomposable.*

Proof. Suppose $\{Q_i\}_{i=1}^m$ is a proper block SOS support of $\Lambda(2k)$. Then for any $Q_i (i = 1, \dots, m)$, we have $\Lambda(k) \not\subseteq Q_i$.

Denote by $\{e_i\}_{i=1}^n$ the unit vectors of $\mathbb{N}^n$. Firstly, we prove there exist $a, b \in \Lambda(k)$ satisfying $a \neq b$, $a - b \in \{-1, 0, 1\}^n$ and $a \notin Q_i \vee b \notin Q_i$ for any $i = 1, \dots, m$.

If every $Q_i (i = 1, ..., m)$ is empty, simply let $a = 0, b = e_1$.

Suppose there exists a non-empty $Q' \in \{Q_i\}_{i=1}^m$. Since $\Lambda(k) \not\subseteq Q'$, there exist $\alpha, \beta \in \Lambda(k)$ such that $\alpha \in Q'$ and $\beta \notin Q'$. Let $\{r_j\}_{j=1}^u \subset \Lambda(k)$ be a path connecting α $(= r_1)$ and β $(= r_u)$ in $\Lambda(k)$ and satisfying for any $j \in \{1, ..., u\}$ there exists $i \in \{1, ..., n\}$ such that $r_{j+1} - r_j = \pm e_i$. Then there exists $j_0 \in \{1, \dots, u-1\}$ such that $r_{j_0} \in Q'$ and $r_{j_0+1} \notin Q'$. We set $a = r_{j_0}, b = r_{j_0+1}$. Because $Q_i (i = 1, \dots, m)$ are pairwise disjoint, $a \notin Q_i \vee b \notin Q_i$ for any $i = 1, \dots, m$.

By Lemma 11, $\{Q_i\}_{i=1}^m$ is also a block SOS support of $\{2a, a + b, 2b\}$. So $(\mathbf{x}^a + \mathbf{x}^b)^2 \in \sum_{i=1}^m \sum \mathbb{R}[\mathbf{x}]_{Q_i}^2$. But by Lemma 10 we see that $(\mathbf{x}^a + \mathbf{x}^b)^2 = \sum_{i=1}^m (\lambda_{i1}\mathbf{x}^a + \lambda_{i2}\mathbf{x}^b)^2$ and there exists i such that $\lambda_{i1}\lambda_{i2} \neq 0$. That contradicts with the claim that for any $i \in \{1, \dots, m\}$, $a \notin Q_i \vee b \notin Q_i$.

So $\Lambda(k)$ is the only block SOS support of $\Lambda(2k)$. ■

Theorem 13 *For any $k \in \mathbb{N}$, the set of block SOS decomposable polynomials in $\mathbb{R}[\mathbf{x}]_{2k}$ has measure zero in $\sum \mathbb{R}[\mathbf{x}]_k^2$.*

Proof. By Theorem 12, we see that the set of block SOS decomposable polynomials in $\mathbb{R}[\mathbf{x}]_{2k}$ is a subset of $\{f \in \mathbb{R}[\mathbf{x}]_{2k} \mid S(f) \neq \Lambda(2k)\}$. So it has measure zero in $\mathbb{R}[\mathbf{x}]_{2k}$. Meanwhile we know $\sum \mathbb{R}[\mathbf{x}]_k^2$ has measure non-zero in $\mathbb{R}[\mathbf{x}]_{2k}$ because the cone $\sum \mathbb{R}[\mathbf{x}]_k^2$ is a full-dimensional convex cone in $\mathbb{R}[\mathbf{x}]_{2k}$, *i.e.*, $\sum \mathbb{R}[\mathbf{x}]_k^2 - \sum \mathbb{R}[\mathbf{x}]_k^2 = \mathbb{R}[\mathbf{x}]_{2k}$. Thus the set of block SOS decomposable polynomials in $\sum \mathbb{R}[\mathbf{x}]_k^2$ must have measure zero in $\sum \mathbb{R}[\mathbf{x}]_k^2$. ■

By the definition of block SOS decomposable polynomial, Theorems 12 and 13 indicate that those polynomials whose SDP matrices (corresponding to their SOS decompositions) can be block-diagonalized are very few in the set of SOS polynomials.

4 Deciding Nonnegativity via `RRC`

In this section we describe in detail the calling sequence, the input and output of `RRC`. We focus on using `RRC` to prove propositions of the form $\phi \Longrightarrow \psi$ where ϕ is a conjunction of polynomial equations, inequations and non-strict inequalities and ψ is a single non-strict polynomial inequality. For other usage of `RRC`, see [24].

4.1 RealRootClassification

To use RRC, first of all, you should install Maple in your computer. The version of Maple should be at least Maple 13. Then, when Maple is started, you should load the RegularChains library as follows before using RRC.

```
> with(RegularChains):
> with(ParametricSystemTools):
> with(SemiAlgebraicSetTools):
```

The calling sequence of RealRootClassification is

$$\texttt{RealRootClassification}(F, N, P, H, d, a, R);$$

where the first four parameter F, N, P and H represent a semi-algebraic system of the following form

$$F = 0, N \geq 0, P > 0, N \neq 0.$$

Herein, each of F, N, P and H is a set of polynomials in unknowns $x_1, ..., x_n$ with rational coefficients. If $F = [f_1, ..., f_s]$, $N = [g_1, ..., g_t]$, $P = [p_1, ..., p_k]$, and $H = [h_1, ..., h_m]$, then $F = 0, N \geq 0, P > 0, N \neq 0$ is a short form for the following system

$$\begin{cases} f_1 = 0, ..., f_s = 0, \\ g_1 \geq 0, ..., g_t \geq 0, \\ p_1 > 0, ..., p_k > 0, \\ h_1 \neq 0, ..., h_m \neq 0. \end{cases}$$

It should be pointed out that s must be positive, *i.e.*, the system must have at least one equation. The last formal parameter R is a list of the variables $x_1, ..., x_n$, which defines an order of the variables and should be defined as a type *PolynomialRing* (see Example 14). The formal parameter d is a positive integer which indicates the last d elements in R are to be viewed as parameters of the given system.

The formal parameter a has two possible forms. If a is a nonnegative integer, then RRC will output the conditions for the system $[F = 0, N \geq 0, P > 0, N \neq 0]$ to have exactly a distinct real solutions. If a is a range, *e.g.* 2..3, then RRC will output the conditions for the number of distinct real solutions of the system $[F = 0, N \geq 0, P > 0, N \neq 0]$ falls into the range a. If the second element of a range is an unassigned name, it means positive infinity.

We illustrate the usage of RRC by the following simple example.

Example 14 *We want to know the conditions on the coefficients of* $f = ax^2 + bx + c$ *for* f *to have real roots if* $a \neq 0$.

After loading RegularChains library and two relative packages, we define the system as follows.

```
> f:=a*x^2+b*x+c;
> F:=[f]; N:=[ ]; P:=[ ]; H:=[a];
> R:=PolynomialRing([x,a,b,c]);
```

To get more information from the output of the function directly, we type in:

```
> infolevel[RegularChains]:=1;
```

Then, we call

```
> RealRootClassification(F, N, P, H, 3, 1..n, R);
```

where the range $1..n$ means "the polynomial has at least one real roots".

The output is: $R_1 > 0$ where $R_1 = b^2 - 4ac$ provided that $a \neq 0$ and $R_1 \neq 0$. To discuss the case when $R_1 = 0$, we can add this equation into the original system and call `RealRootClassification` again.

```
> RealRootClassification([b^2-4*a*c,op(F)], N, P, H, 3, 1..n,R);
```

In this way, we finally know that the condition is $R_1 \geq 0$.

4.2 More Examples by RRC

We first give a detailed explanation of Example 2. In Example 2, we call

```
> RealRootClassification([abc-1], [a, b, c], [-f], [ ], 2, 0, R);
```

where the "0" means we want to compute the conditions for the system to have no real solutions.

The output is:

There is always given number of real solution(s)!
PROVIDED THAT
$\phi(b,c) \neq 0,$

where $\phi(b,c)$ is a polynomial in b and c with 19 terms and of degree 18.

The output means that the system always has no real solutions provided that the polynomial $\phi(b,c)$ does not vanish. In other word, RRC proves that the proposition holds for almost all a, b and c except those such that $\phi(b,c) = 0$.

Because the inequality to be proved is a non-strict inequality ($f \geq 0$), by continuity, we know at once that $f \geq 0$ holds for all a, b and c such that $a \geq 0 \wedge b \geq 0 \wedge c \geq 0 \wedge abc - 1 = 0$. Thus, the proposition is proved.

Example 15 *Prove that*

$$a \geq 0 \wedge b \geq 0 \wedge c \geq 0 \wedge ab + bc + ca - 1 = 0 \Longrightarrow g \geq 0$$

where

$$\begin{aligned} g = \ & -10a^3b^3 - 10b^3c^3 - 10a^3c^3 - 5a^4b^2 - 5c^2a^4 - 5c^4a^2 - 5a^2b^4 + 4c^3a \\ & -5b^4c^2 - 5b^2c^4 + 4ca^3 + 2a^4 + 2b^4 + 2c^4 - 10cab^4 - 30c^2a^3b - 10ca^4b \\ & -10c^4ab + 4a^3b^4c + 16a^3b^3c^2 + 4a^4b^3c + 16b^3c^3a^2 + 16a^3c^3b^2 \\ & +4a^3c^4b + 4b^3c^4a + 4b^4c^3a + 4a^4c^3b + 6b^2c^2a^4 - 30b^3c^2a \\ & -30c^3a^2b + 6b^2c^4a^2 + 16c^2ab + 16ca^2b - 50b^2c^2a^2 + 16cab^2 - 30b^2c^3a \\ & -30ca^3b^2 - 30a^2b^3c + 6b^4c^2a^2 + 6c^2a^2 + 6a^2b^2 + 6b^2c^2 + 4c^3b + 4b^3c \\ & +4b^3a + 4a^3b + 2a^4b^4 + 2a^4c^4 + 2b^4c^4. \end{aligned}$$

Example 16 *Prove that*

$$x \geq 0 \wedge y \geq 0 \wedge z \geq 0 \wedge r \geq 0 \wedge (r+1)^2 - 4/3 \geq 0 \wedge x + y + z - 3 = 0 \Longrightarrow h \geq 0$$

where

$$\begin{aligned} h = \ & -3 + z - 3r^3y^2z^2x^2 + ry^3 + r^2z^3 + rz^3 - 3ry^2 - 3rz^2 + r^2x^3 + yr \\ & +r^2y^3 + zr + rx^3 + rx - 3rx^2 + xr^2z^2 + yrx^2 + xrz^2 + r^3y^3x^2 \\ & +r^2y^3x^2 - 3r^2y^2z^2 + r^2y^2z^3 + r^3z^2x^3 + r^2z^2x^3 + zr^2y^2 + zry^2 \\ & +yr^2x^2 - 3r^2z^2x^2 - 3r^2y^2x^2 + r^3y^2z^3 + y + x. \end{aligned}$$

Example 17 *Prove that*

$$a \geq 0 \wedge b \geq 0 \wedge c \geq 0 \wedge d \geq 0 \wedge a+b+c+d-1=0 \Longrightarrow p \geq 0$$

where $p = 1 + 176abcd - 27(bcd + cda + dab + abc)$.

Example 18 *Prove that for any given integer* $n \geq 3$,

$$-1 \leq x_i \leq 1 \ (1 \leq i \leq n) \wedge \sum x_i^3 = 0 \Longrightarrow \sum x_i \leq \frac{n}{3}.$$

Although the problem is not so hard for a mathematician, it is really hard for a computer. We proved the proposition for $n = 3, 4, 5$ *by* `RRC`.

Example 19 *Prove that*

$$a \geq 0 \wedge b \geq 0 \wedge c \geq 0 \wedge a^3b + b^3c + c^3a - 3 = 0 \Longrightarrow q \geq 0$$

where $q = -75a^4b^4c^4 - 5a^4b^4 - 5a^4c^4 - 5b^4c^4 + 21a^4 + 21b^4 + 21c^4 + 27$.

Example 20 [1] *Prove that*

$$a \geq 0 \wedge b \geq 0 \wedge c \geq 0 \wedge a^3b + ac^3 + b^3c + abc - 4 = 0 \Longrightarrow w \geq 0$$

where $w = 27(a+b+c)^4 - 1024$.

Examples 15-20 have a common property that the systems themselves have at least one equation. So, we can use `RRC` directly. We show by the following two examples how to deal with the situation where no equations appear in the system.

Example 21 *Prove that*

$$a \geq 0 \wedge b \geq 0 \wedge c \geq 0 \wedge d \geq 0 \Longrightarrow u \geq 0$$

where

$$\begin{aligned} u = \ & 1280bd^3c + 624bc^2d^2 + 320ab^4 + 464ac^4 - 112ad^4 - 112a^4b + 464a^4c \\ & -112b^4c + 464b^4d + 208c^3b^2 + 1072d^3b^2 - 224b^3c^2 + 1072b^3d^2 \\ & +320bc^4 + 464bd^4 - 112c^4d + 208d^3c^2 - 224c^3d^2 + 320cd^4 + 128ad^3c \\ & +624ab^2c^2 + 740b^3cd + 1812ab^2d^2 + 516ac^2d^2 + 1812b^2cd^2 \\ & +128bc^3d + 516b^2c^2d + 128a^3bd + 624a^2b^2d + 516a^2bd^2 + 1280a^3cd \\ & +1812a^2c^2d + 624a^2cd^2 + 128ab^3c + 1280ab^3d + 1280ac^3b + 740ac^3d \\ & +740ad^3b + 1812a^2bc^2 + 740a^3bc + 516a^2b^2c + 1896ab^2cd + 1896abc^2d \\ & +1896abcd^2 + 1896a^2bcd + 320a^4d + 208b^3a^2 + 1072c^3a^2 - 224d^3a^2 \\ & -224a^3b^2 + 1072a^3c^2 + 208a^3d^2 + 64a^5 + 64b^5 + 64c^5 + 64d^5. \end{aligned}$$

As usual, we want to prove that the following system has no real solutions

$$a \geq 0 \wedge b \geq 0 \wedge c \geq 0 \wedge d \geq 0 \wedge u < 0.$$

[1] http://www.artofproblemsolving.com/Forum/viewtopic.php?f=52&t=432676

However, the system does not contain equations and thus RRC *cannot be applied directly.*

We introduce a new variable T and the system being inconsistent is equivalent to that the following new system is inconsistent

$$a \geq 0 \wedge b \geq 0 \wedge c \geq 0 \wedge d \geq 0 \wedge u + T = 0 \wedge T > 0.$$

For this new problem, we first define
```
> R := PolynomialRing([T, a, b, c, d]);
```
and then call
```
> RealRootClassification([u+T], [a, b, c, d], [T], [ ], 4, 0, R);
```
The problem is solved immediately.

Example 22 *Prove that*

$$a \geq 0 \wedge b \geq 0 \wedge c \geq 0 \Longrightarrow v \geq 0$$

where

$$\begin{aligned} v = \ & 104976a^{12} + 1679616a^{11}b + 1469664a^{11}c + 10850112a^{10}b^2 \\ & +19046016a^{10}bc + 8076024a^{10}c^2 + 36149760a^9b^3 + 95364864a^9b^2c \\ & +80561952a^9bc^2 + 22935528a^9c^3 + 65762656a^8b^4 + 228601856a^8b^3c \\ & +282635520a^8b^2c^2 + 162625040a^8bc^3 + 42710593a^8c^4 + 63474176a^7b^5 \\ & +251921856a^7b^4c + 354740704a^7b^3c^2 + 288770224a^7b^2c^3 \\ & +207550776a^7bc^4 + 83017484a^7c^5 + 29076288a^6b^6 + 60534016a^6b^5c \\ & -155234320a^6b^4c^2 - 380047056a^6b^3c^3 + 3130676a^6b^2c^4 \\ & +375984436a^6bc^5 + 181119606a^6c^6 + 8313344a^5b^7 - 89738240a^5b^6c \\ & -760459488a^5b^5c^2 - 1768157568a^5b^4c^3 - 1403613720a^5b^3c^4 \\ & +236428572a^5b^2c^5 + 824797636a^5bc^6 + 291288188a^5c^7 \\ & +13943056a^4b^8 - 3628032a^4b^7c - 514131904a^4b^6c^2 - 1869896304a^4b^5c^3 \\ & -2495402586a^4b^4c^4 - 783163260a^4b^3c^5 + 1171287578a^4b^2c^6 \\ & +1122586500a^4bc^7 + 288706561a^4c^8 + 18028800a^3b^9 + 116005472a^3b^8c \\ & +171678496a^3b^7c^2 - 347011440a^3b^6c^3 - 1231272792a^3b^5c^4 \\ & -894635820a^3b^4c^5 + 731754984a^3b^3c^6 + 1497257080a^3b^2c^7 \\ & +851454308a^3bc^8 + 170469720a^3c^9 + 10593792a^2b^{10} + 100409472a^2b^9c \\ & +365510616a^2b^8c^2 + 624203728a^2b^7c^3 + 480156788a^2b^6c^4 \\ & +215762988a^2b^5c^5 + 511667522a^2b^4c^6 + 990571720a^2b^3c^7 \\ & +861820134a^2b^2c^8 + 356931720a^2bc^9 + 58375800a^2c^{10} \\ & +2985984ab^{11} + 34730496ab^{10}c + 165207744ab^9c^2 + 415788248ab^8c^3 \\ & +606389880ab^7c^4 + 560561092ab^6c^5 + 437187748ab^5c^6 + 422470380ab^4c^7 \\ & +390424292ab^3c^8 + 235263240ab^2c^9 + 77497200abc^{10} + 10692000ac^{11} \\ & +331776b^{12} + 4478976b^{11}c + 25292160b^{10}c^2 + 77899104b^9c^3 \\ & +144247489b^8c^4 + 170606684b^7c^5 + 141892350b^6c^6 + 102086036b^5c^7 \\ & +76748161b^4c^8 + 52182360b^3c^9 + 24766200b^2c^{10} + 6804000bc^{11} \\ & +810000c^{12}. \end{aligned}$$

Similar to Example 21, the inequality is proved by first defining
```
> R := PolynomialRing([T, a, b, c]);
```
and then calling
```
> RealRootClassification([v+T], [a, b, c], [T], [ ], 3, 0, R);
```

The timings of RRC on Examples 15-22 are listed in the following table. All the computation were performed on a computer (CPU 3.2GHz, 2G RAM, Windows XP) with Maple 13.

Timings (in second) of RRC on Examples 15-22.

#	15	16	17	18($n = 5$)	19	20	21	22
timings	0.04	6.04	0.03	377.35	16.67	2.98	1.26	0.57

5 Conclusions

SparseSOS is a very efficient tool for detecting global nonnegativity of polynomials and RRC in Maple can be used to prove nonnegativity of polynomials under semi-algebraic constraints. The usage of the two tools are described in detail with many examples. In order to discuss on the question how many polynomials can the existing sparse SOS decomposition algorithms be applied to, we define those polynomials as *block SOS decomposable polynomials* and prove that the set of block SOS decomposable polynomials is measure zero in the set of SOS polynomials.

References

[1] Blekherman, G., There are significantly more nonegative polynomials than sums of squares, *Israel Journal of Mathematics*, **153**, 1, 355–380, 2006.

[2] Chen, C., Davenport, J. H., Lemaire, F., Maza, M. M., Xia, B., Xiao, R. and Xie, Y., Computing the real solutions of polynomial systems with the RegularChains library in Maple, *Software Demo at ISSAC 2011, Communications in Computer Algebra*, **45**, 3/4, 2012.

[3] Choi, M.-D., Lam, T. Y. and Reznick, B., Sums of squares of real polynomials, *Proceedings of Symposia in Pure Mathematics*, Vol. 58 (American Mathematical Society), 103–126, 1995.

[4] Collins, G. E., Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition, *LNCS 33*, 134–183, Springer-Verlag, Berlin, 1975.

[5] Dai, L. and Xia, B., Smaller SDP for SOS decomposition, *Journal of Global Optimization*, **63**, 2, 343–361, 2015.

[6] Kaltofen, E., Li, B., Yang, Z. and Zhi, L., Exact certification of global optimality of approximate factorizations via rationalizing sums-of-squares with floating point scalars, *Proc. ISSAC'2008* (ACM Press), 155–164, 2008.

[7] Kim, S., Kojima, M. and Waki, H., Generalized lagrangian duals and sums of squares relaxations of sparse polynomial optimization problems, *SIAM Journal on Optimization*, **15**, 3, 697–719, 2005.

[8] Lasserre, J., Global optimization with polynomials and the problem of moments, *SIAM Journal on Optimization* **11**, 3, 796–817, 2001.

[9] Li, H. and Xia, B., Block SOS Decomposition, *arXiv:1801.07954*, 2018.

[10] Lofberg, J., Yalmip: A toolbox for modeling and optimization in matlab, *2004 IEEE International Symposium on Computer Aided Control Systems Design*, 284–289, 2004.

[11] Löfberg J., Pre- and Post-Processing Sum-of-Squares Programs in Practice, *IEEE Transactions on Automatic Control*, **54** 5, 1007–1011, 2009.

[12] Papachristodoulou, A., Anderson, J., Valmorbida, G., Prajna, S., Seiler, P. and Parrilo, P. A., SOSTOOLS: Sum of squares optimization toolbox for MATLAB, *arXiv:1310.4716*, 2013.

[13] Parrilo, P. A., *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*, Ph.D. Thesis, California Institute of Technology, 2000.

[14] Permenter, F. and Parrilo, P. A., Finding sparse, equivalent SDPs using minimal coordinate projections, *Proc. 54th IEEE Conference on Decision and Control*, 7274–7279, 2015.

[15] Powers, V. and Wörmann, T., An algorithm for sums of squares of real polynomials, *Journal of pure and applied algebra* **127**, 1, 99–104, 1998.

[16] Reznick, B., Extremal PSD forms with few terms, *Duke Math. J.*, **45**, 363–374, 1978.

[17] Schweighofer, M., Optimization of polynomials on compact semialgebraic sets, *SIAM Journal on Optimization* **15**, 3, pp. 805–825, 2005.

[18] Seiler, P., Sosopt: A toolbox for polynomial optimization, *arXiv:1308.1889*, 2013.

[19] Tarski, A., *A Decision Method for Elementary Algebra and Geometry*, University of California Press, Berkley, 1951.

[20] Vandenberghe, L. and Boyd, S. (1996). Semidefinite programming, *SIAM Review* **38**, 1, pp. 49–95.

[21] Waki, H., Kim, S., Kojima, M., Muramatsu, M. and Sugimoto, H., Algorithm 883: SparsePOP—A Sparse Semidefinite Programming Relaxation of Polynomial Optimization Problems, *ACM Transactions on Mathematical Software (TOMS)*, 2008, **35**, 2, 15.

[22] Wang, J., Li, H. and Xia, B., A New Sparse SOS Decomposition Algorithm Based on Term Sparsity, *Proc. ISSAC 2019*, 347–354, 2019.

[23] Xia, B., DISCOVERER: a tool for solving semi-algebraic systems, *ACM Commun. Comput. Algebra*, **41**, 3, 102–103, 2007.

[24] Xia, B. and Yang, L.: *Automated inequality proving and discovering*, World Scientific, Singapore, 2016.

[25] Yang, L., Hou, X. and Xia, B., A complete algorithm for automated discovering of a class of inequality-type theorems, *Sci. China, Series* **F**, **44**, 6, 33–49, 2001.

[26] Yang, L., Hou, X. and Zeng, Z., A Complete Discrimination System for Polynomials, *Sci. China*, Series E, **39**, 6, 628–646, 1996.

[27] Yang, L. and Xia, B., Real solution classifications of a class of parameteric semi-algebraic systems, *A3L 2005*, 281–289, Herstellung and Verlag, Norderstedt, 2005.

[28] Yang, L. and Xia, B., Deciding nonnegativity of polynomials by MAPLE, arXiv:1306.4059.