RealCertify: a Maple package for certifying non-negativity

Victor Magron
CNRS Verimag, Sorbonne Université, INRIA,
Laboratoire d'Informatique de Paris 6, LIP6, Équipe PolSys
victor.magron@lip6.fr

Mohab Safey El Din
Sorbonne Université, CNRS, INRIA,
Laboratoire d'Informatique de Paris 6, LIP6, Équipe PolSys
mohab.safey@lip6.fr

Abstract

Let $\mathbb{Q}$ (resp. $\mathbb{R}$) be the field of rational (resp. real) numbers and $X = (X_1, \ldots, X_n)$ be variables. Deciding the non-negativity of polynomials in $\mathbb{Q}[X]$ over $\mathbb{R}^n$ or over semi-algebraic domains defined by polynomial constraints in $\mathbb{Q}[X]$ is a classical algorithmic problem for symbolic computation.

The Maple package RealCertify tackles this decision problem by computing sum of squares certificates of non-negativity for inputs where such certificates hold over the rational numbers. It can be applied to numerous problems coming from engineering sciences, program verification and cyber-physical systems. It is based on hybrid symbolic-numeric algorithms based on semi-definite programming.

1 Introduction

Let $\mathbb{Q}$ (resp. $\mathbb{R}$) be the field of rational (resp. real) numbers and $X = (X_1, \ldots, X_n)$ be a sequence of variables. We consider the problem of deciding the non-negativity of $f \in \mathbb{Q}[X]$ either over $\mathbb{R}^n$ or over a semi-algebraic set S defined by some constraints $g_1 \geq 0, \ldots, g_m \geq 0$ (with $g_j \in \mathbb{Q}[X]$). We denote by d the maximum of the total degrees of these polynomials.

The Cylindrical Algebraic Decomposition (CAD) algorithm [2] solves this decision problem in time doubly exponential in n (and polynomial in d). This algorithm (and its further improvements) has been implemented in most of computer algebra systems.

Later, the so-called critical point method has been designed, allowing to solve this decision problem in time singly exponential in n (and polynomial in d). Recent variants of this method have also been implemented in the RAGLib Maple package.

All the aforementioned algorithms are "root finding" ones: they try to find a point at which f is negative over the considered domain. When f is positive, they return an empty list without a *certificate* that can be checked *a posteriori*.

To compute certificates of non-negativity, an approach based on *sum of squares* (SOS) decompositions (and their variants) has been popularized by Lasserre [4] and Parillo [8]. The idea is as follows.

To ensure that a polynomial f of degree $d = 2k$ is non-negative over $\mathbb{R}^n$, it suffices to write it as a sum of squares $c_1 s_1^2 + \cdots + c_r s_r^2$ where the c_i's are positive constants. When such s_i and such c_i's can be obtained with rational coefficients, one says that one obtains a certificate of non-negativity over the rationals. Such a decomposition can be obtained by finding a semi-definite positive symmetric matrix G such that

$$f = v_k^T G v_k$$

where v_k is the vector of all monomials of degree $\leq k$. Obtaining such a matrix G boils down to solving a linear matrix inequality.

This method is attractive because efficient numerical solvers are available for solving *large* linear matrix inequalities. Besides, when d is fixed and n grows, the size of the matrix G varies polynomially in n, hence providing *approximations* of a sum of squares decomposition for f. It can also be generalized to obtain certificates of non-negativity for constrained problems, writing f as

$$f = \sigma_0 + \sigma_1 g_1 + \cdots + \sigma_m g_m$$

where the σ_i's are sum of squares.

On the minus side, this method provides only *approximations* of certificates of non-negativity. Besides, it is well-known that not all non-negative polynomials can be written as sum of squares of polynomials. Original work of Parillo/Peyrl [9] and Kaltofen/Li/Yang/Zhi [3] have opened the door to hybrid symbolic numeric strategies for computing certificates of non-negativity whenever such certificates exist over the rational numbers.

In [5], we have designed hybrid symbolic-numeric algorithms for computing certificates of non-negativity over the rationals in some "easy" situations (roughly speaking, these are the situations where the searched sum of squares decomposition lies in the interior of the cone of polynomials which are sum of squares). The package RealCertify implements these algorithms and aims at providing a full suite of hybrid algorithms for computing certificates of non-negativity based on numerical software for solving linear matrix inequalities.

2 Algorithmic background and overall description

2.1 The univariate case

In the univariate case, all non-negative polynomials are sums-of-squares. The library includes two distinct algorithms:

- `univsos1`, which is a recursive procedure relying on root isolation and quadratic under approximations of positive polynomials. The first step computes a rational approximation t of the smallest global minimizer a of f and a non-negative quadratic under-approximation f_t of f such that t is a root of $f - f_t$. The second step performs square-free decomposition of $f - f_t = gh^2$. Then, we apply the same procedure on g until the resulting degree is less than 2.

- `univsos2`, which relies on root isolation of perturbed positive polynomials. Given a univariate polynomial $f > 0$ of degree $d = 2k$, this algorithm computes weighted SOS decompositions of f. The first numeric step of `univsos2` is to find ε such that the perturbed polynomial $f_\varepsilon := f - \varepsilon \sum_{i=0}^{k} X^{2i} > 0$ and to compute its complex roots, yielding an approximate SOS decomposition $l(s_1^2 + s_2^2)$, where l is the leading coefficient of f_ε. In the second symbolic step, one considers the remainder polynomial $u := f_\varepsilon - l s_1^2 - l s_2^2$ and tries to computes an exact SOS decomposition of $\varepsilon \sum_{i=0}^{k} X^{2i} + u$. This succeeds for large enough precision of the root isolation procedure.

In both cases, the output is a list $[c_1, s_1, \ldots, c_r, s_r]$, with $c_i \in \mathbb{Q}^{>0}$, $s_i \in \mathbb{Q}[X]$, such that $f = c_1 s_1^2 + \cdots + c_r s_r^2$. Let us illustrate the behavior of both algorithms on the input $f = 1 + X + X^2 + X^3 + X^4$.

1. When running `univsos1`, the algorithm first provides the value $t = -1$ as an approximation of the minimizer of f together with a positive quadratic under-approximation $f_t(X) = X^2$. Next, one obtains the square-free decomposition $f - f_t = (X+1)^2 g(X)$ with $g(X) = (X - \frac{1}{2})^2 + \frac{3}{4}$. The Maple command: `univsos1(1+X+X^2+X^3+X^4,X)` outputs the list $[1, 0, 1, (X+1)(X - \frac{1}{2}), \frac{3}{4}, X+1, 1, -X]$, corresponding to the weighted rational SOS decomposition $f = (X+1)^2(X-\frac{1}{2})^2 + \frac{3}{4}(X+1)^2 + (-X)^2$.

2. When running `univsos2`, the algorithm performs the first loop and provides the value $\varepsilon = \frac{1}{8}$ with the polynomial $f_\varepsilon := f - \frac{1}{8}(1 + X^2 + X^4)$ which has no real root. The leading coefficient of f_ε is $l = \frac{7}{8}$. After multiplying the precision of complex root isolation by 8, one obtains $s_1 = X^2 + \frac{9}{16}X - \frac{3}{4}$, $s_2 = \frac{23}{16}X + \frac{11}{16}$ and $u = \frac{1}{64}X^3 + \frac{105}{1024}X^2 + \frac{9}{1024}X - \frac{63}{2048}$. Using that $X^3 = \frac{1}{2}(X^2 + X)^2 - \frac{1}{2}(X^4 + X^2)$ and $X = \frac{1}{2}(X + 1)^2 - \frac{1}{2}(X^2 + 1)$, one gets an SOS decomposition for $u + \frac{1}{8}(1 + X^2 + X^4)$. The Maple command `univsos2(1+X+X^2+X^3+X^4,X)` outputs the decomposition $f = \frac{7}{8}(s_1^2 + s_2^2) + \frac{377}{4096} + \frac{55}{256}X^2 + \frac{7}{64}X^4 + \frac{9}{1024}(X + \frac{1}{2})^2 + \frac{1}{64}X^2(X + \frac{1}{2})^2$.

2.2 The multivariate case

In the multivariate case, the `multivsos` library performs SOS decompositions of multivariate non-negative polynomials with rational coefficients in the (un)-constrained case.

In the unconstrained case, `multivsos` implements a hybrid numeric-symbolic algorithm computing exact rational SOS decompositions for polynomials lying in the interior $\mathring{\Sigma}[X]$ of the SOS cone $\Sigma[X]$. It computes an approximate SOS decomposition for a perturbation of the input polynomial with an arbitrary-precision semi-definite programming (SDP) solver. An exact SOS decomposition is obtained thanks to the perturbation terms. Given $f \in \mathbb{Z}[X] \cap \mathring{\Sigma}[X]$ of degree $d = 2k$, one first computes its Newton polytope P. The support of the SOS involved in the decomposition of f lies in $Q = P/2 \cap \mathbb{N}^n$. A first loop allows to find $\varepsilon \in \mathbb{Q}^{>0}$ such that the perturbed polynomial $f_\varepsilon := f - \varepsilon \sum_{\alpha \in Q} X^{2\alpha}$ is also in $\mathbb{Z}[X] \cap \mathring{\Sigma}[X]$. In the second loop, one computes an approximate rational SOS decomposition $\tilde{\sigma}$ of f_ε with an arbitrary-precision SDP solver (`sdp` procedure). We obtain the remainder $u = f - \varepsilon \sum_{\alpha \in Q} X^{2\alpha} - \tilde{\sigma}$. When the precision is large enough, the last symbolic step allows to retrieve an exact rational SOS decomposition of $u + \varepsilon \sum_{\alpha \in Q} X^{2\alpha}$.

In the constrained case, `multivsos` relies on a similar procedure to compute weighted SOS decompositions for polynomials positive over basic compact semi-algebraic sets.

We apply `multivsos` on $f = 4X_1^4 + 4X_1^3X_2 - 7X_1^2X_2^2 - 2X_1X_2^3 + 10X_2^4$. The other input parameters are $\varepsilon = 1$, $\delta = R = 60$ and $\delta_c = 10$. Then $Q := \operatorname{conv}(\operatorname{spt}(f))/2 \cap \mathbb{N}^n = \{(2,0), (1,1), (0,2)\}$. At the end of the first loop, we get $f - \varepsilon t = f - (X_1^4 + X_1^2X_2^2 + X_2^2) \in \mathring{\Sigma}[X]$. The `sdp` and `cholesky` procedures yield $s_1 = 2X_1^2 + X_1X_2 - \frac{8}{3}X_2^2$, $s_2 = \frac{4}{3}X_1X_2 + \frac{3}{2}X_2^2$ and $s_3 = \frac{2}{7}X_2^2$. The remainder polynomial is $u = f - \varepsilon t - s_1^2 - s_2^2 - s_3^2 = -X_1^4 - \frac{1}{9}X_1^2X_2^2 - \frac{2}{3}X_1X_2^3 - \frac{781}{1764}X_2^4$.

At the end of the second loop, we obtain $\varepsilon_{(2,0)} = \varepsilon - X_1^4 = 0$, which is the coefficient of X_1^4 in $\varepsilon t + u$. Then, $\varepsilon(X_1^2X_2^2 + X_2^4) - \frac{2}{3}X_1X_2^3 = \frac{1}{3}(X_1X_2 - X_2^2)^2 + (\varepsilon - \frac{1}{3})(X_1^2X_2^2 + X_2^4)$. In the polynomial $\varepsilon t + u$, the coefficient of $X_1^2X_2^2$ is $\varepsilon_{(1,1)} = \varepsilon - \frac{1}{3} - \frac{1}{9} = \frac{5}{9}$ and the coefficient of X_4^4 is $\varepsilon_{(0,2)} = \varepsilon - \frac{1}{3} - \frac{781}{1764} = \frac{395}{1764}$.

The Maple command

```
multivsos(4 * X1^4 + 4 * X1^3 * X2 - 7 * X1^2 * X2^2 - 2 * X1 * X2^3 + 10 * X2^4):
```

allows to obtain the weighted rational SOS decomposition: $4X_1^4 + 4X_1^3X_2 - 7X_1^2X_2^2 - 2X_1X_2^3 + 10X_2^4 = \frac{1}{3}(X_1X_2 - X_2^2)^2 + \frac{5}{9}(X_1X_2)^2 + \frac{395}{1764}X_2^4 + (2X_1^2 + X_1X_2 - \frac{8}{3}X_2^2)^2 + (\frac{4}{3}X_1X_2 + \frac{3}{2}X_2^2)^2 + (\frac{2}{7}X_2^2)^2$.

2.3 Dependencies

The RealCertify software is available and maintained as a GitLab repository at RealCertify. The `univsos` and `multivsos` libraries have been tested with Maple 2016. `univsos` requires the external PARIGP software for `univsos2`, as well as the external SDP solvers SDPA (double precision) and SDPA-GMP [7] (arbitrary-precision). In addition of SDPA and SDPA-GMP used for the `sdp` procedure, `multivsos` requires the Maple package Convex, by M. Franz, to compute Newton polytopes.

3 Performance analysis and limitations

Timings, which we report on below, were obtained on an Intel Core i7-5600U CPU (2.60 GHz) with 16Gb of RAM. Most of the time is often spent in the `sdp` procedure for all benchmarks. Those benchmarks are standard ones in the polynomial optimization community. We report here only on multivariate problems (see the references in [5]). We refer to [6] for a performance analysis in the univariate case. The table on the left below reports on unconstrained problems, while the one on the right reports on constrained ones ("–" means that a decomposition cannot be obtained within a day of computation).

Id	n	d	multivsos		RAGLib	CAD
			output bitsize	time (s) total/sdp	time (s)	time (s)
f_{12}	2	12	162 861	5.96/5.20	0.15	0.07
f_{20}	2	20	745 419	110./106.	0.16	0.03
M_{20}	3	8	4 695	0.18/0.01	0.13	0.05
M_{100}	3	8	17 232	0.35/0.05	0.15	0.03
r_2	2	4	1 866	0.03/0.01	0.09	0.01
r_4	4	4	14 571	0.15/0.01	0.32	–
r_6	6	4	56 890	0.34/0.03	623.	–
r_8	8	4	157 583	0.96/0.09	–	–
r_{10}	10	4	344 347	2.45/0.72	–	–
r_6^2	6	8	1 283 982	13.8/10.2	10.9	–

Id	n	d	multivsos		RAGLib	CAD
			output bitsize	time (s) total/sdp	time (s)	time (s)
p_{46}	2	4	21 723	0.83/0.29	0.15	0.81
f_{260}	6	3	114 642	2.72/1.94	0.12	–
f_{491}	6	3	108 359	9.65/4.46	0.01	0.05
f_{752}	6	2	10 204	0.26/0.01	0.07	–
f_{859}	6	7	6 355 724	303./259.	5896.	–
f_{863}	4	2	5 492	0.14/0.01	0.01	0.01
f_{884}	4	4	300 784	25.1/25.1	0.21	–
f_{890}	4	4	60 787	0.59/0.18	0.08	–
butcher	6	3	247 623	1.32/0.21	47.2	–
heart	8	4	618 847	2.94/1.13	0.54	–
magn.	7	2	9 622	0.29/0.02	434.	–

Some conclusions can already be drawn from our experiments.

1. There is a *wide* class of semi-algebraic problems for which emptiness can be *proven* with exact certificates using RealCertify. On this class, RealCertify outperforms the current state-of-the-art. One striking fact is that, for most of these problems, certification is not an issue and is negligible compared to the time spent by SDP solvers.

   Mathematically, this class is restricted to unconstrained problems involving SOS polynomials lying in the interior of the SOS cone or constrained problems defining basic compact semi-algebraic sets.

2. However, we attract the attention of the reader to the fact that the class of unconstrained problems on which RealCertify cannot be applied is wide too. Also, even under our mathematical assumptions, we met some examples that can be tackled by the state-of-the-art but not by RealCertify. The bottleneck is currently the SDP solving step with high-precision.

All in all, RealCertify outperforms the state-of-the-art on some class of problems *and* provides exact certificates of inconsistency, while it is not as general as other tools. Hence, it is a useful and powerful complement to them.

References

[1] S. Chevillard, J. Harrison, M. Joldes, and C. Lauter. Efficient and accurate computation of upper bounds of approximation errors. *Theoretical Computer Science*, 412(16):1523 – 1543, 2011.

[2] G. E Collins. Quantifier elimination for real closed fields by cylindrical algebraic decompostion. In *ATFL 2nd GI Conf. Kaiserslautern*, pages 134–183, 1975.

[3] E. Kaltofen, B. Li, Z. Yang, and L. Zhi. Exact certification of global optimality of approximate factorizations via rationalizing sums-of-squares with floating point scalars. In *Proceedings of ISSAC*, pp. 155–164. ACM, 2008.

[4] J.-B. Lasserre. Global Optimization with Polynomials and the Problem of Moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[5] V. Magron and M. Safey El Din. On Exact Polya and Putinar's Representations. To appear in *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation (ISSAC)*.

[6] V. Magron, M. Safey El Din, and M. Schweighofer. Algorithms for Weighted Sum of Squares Decomposition of Non-negative Univariate Polynomials. *Journal of Symbolic Computation*, doi.org/10.1016/j.jsc.2018.06.005.

[7] M. Nakata. A numerical evaluation of highly accurate multiple-precision arithmetic version of semidefinite programming solver: SDPA-GMP, -QD and -DD. In *CACSD*, pages 29–34, 2010.

[8] P. A. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization.* PhD thesis, California Inst. Tech., 2000.

[9] H. Peyrl and P.A. Parrilo. Computing sum of squares decompositions with rational coefficients. *Theoretical Computer Science*, 409(2):269–281, 2008.