Verifying nonlinear real formulas via sums of squares

John Harrison

Intel Corporation, JF1-13
2111 NE 25th Avenue, Hillsboro OR 97124, USA
johnh@ichips.intel.com

Abstract. Techniques based on sums of squares appear promising as a general approach to the universal theory of reals with addition and multiplication, i.e. verifying Boolean combinations of equations and inequalities. A particularly attractive feature is that suitable 'sum of squares' certificates can be found by sophisticated numerical methods such as semidefinite programming, yet the actual verification of the resulting proof is straightforward even in a highly foundational theorem prover. We will describe our experience with an implementation in HOL Light, noting some successes as well as difficulties. We also describe a new approach to the univariate case that can handle some otherwise difficult examples.

1 Verifying nonlinear formulas over the reals

Over the real numbers, there are algorithms that can in principle perform quantifier elimination from arbitrary first-order formulas built up using addition, multiplication and the usual equality and inequality predicates. A classic example of such a quantifier elimination equivalence is the criterion for a quadratic equation to have a real root:

$$\forall a\ b\ c.\ (\exists x.\ ax^2 + bx + c = 0) \Leftrightarrow a = 0 \land (b = 0 \Rightarrow c = 0) \lor a \neq 0 \land b^2 \geq 4ac$$

The first quantifier elimination algorithm for this theory was developed by Tarski [32],[1] who actually demonstrated completeness and quantifier elimination just for the theory of real-closed fields, which can be characterized as ordered fields where all nonnegative elements have square roots ($\forall x.\ 0 \leq x \Rightarrow \exists y.\ x = y^2$) and all non-trivial polynomials of odd degree have a root. There are several interesting models of these axioms besides the reals (e.g. the algebraic reals, the computable reals, the hyperreals) yet Tarski's result shows that these different models satisfy exactly the same properties in the first-order language under consideration.

However, Tarski's procedure is complicated and inefficient. Many alternative decision methods were subsequently proposed; two that are significantly simpler were given by Seidenberg [30] and Cohen [8], while the CAD algorithm [9], apparently the first ever to be implemented, is significantly more efficient, though relatively complicated. Cohen's ideas were recast by Hörmander [17] into a relatively simple algorithm. However, even CAD has poor worst-case complexity (doubly exponential), and the Cohen-Hörmander algorithm is generally still slower. Thus, there has been limited progress on

[1] Tarski actually discovered the procedure in 1930, but it remained unpublished for many years afterwards.

applying these algorithms to problems of interest. An interesting alternative, currently unimplemented, is described in [4].

If we turn to implementation in foundational theorem provers using basic logical operations instead of complicated code, the situation is bleaker still. The Cohen-Hörmander algorithm has been implemented in Coq [23] and HOL Light [24] in a way that generates formal proofs. However, producing formal proofs induces a further significant slowdown. In practice, more successful approaches to nonlinear arithmetic tend to use heuristic approaches that work well for simple common cases like $x > 0 \land y > 0 \Rightarrow xy > 0$ but are incomplete (or at least impractical) in general [18, 33], though in some cases they go beyond the simple algebraic operations [1].

But many important problems in practice are purely universally quantified, i.e. are of the form $\forall x_1, \ldots, x_n. P[x_1, \ldots, x_n]$ where $P[x_1, \ldots, x_n]$ is an arbitrary Boolean combination of polynomial equations and inequalities. Typical (true) examples are $\forall x\ y. x \leq y \Rightarrow x^3 \leq y^3$, $\forall x. 0 \leq 1 \land x \leq 1 \Rightarrow x^2 \leq 1$ and $\forall a\ b\ c\ x. ax^2 + bx + c = 0 \Rightarrow b^2 \geq 4ac$. For this logically restrictive but practically important case, a completely different approach is possible based on *sums of squares*.

2 Positivity and sums of squares

We will be concerned with the set of multivariate polynomials $\mathbb{R}[x_1, \ldots, x_n]$ over the reals, and often more specifically the subset $\mathbb{Q}[x_1, \ldots, x_n]$ with rational coefficients. The cornerstone of what follows is the relationship between a polynomial's taking nonnegative values everywhere, a.k.a. being *positive semidefinite* (PSD):

$$\forall x_1, \ldots, x_n.\, p(x_1, \ldots, x_n) \geq 0$$

and the existence of a decomposition into a sum of squares (SOS) of other polynomials:

$$p(x_1, \ldots, x_n) = \sum_{i=0}^{k} s_i(x_1, \ldots, x_n)^2$$

Since any square is nonnegative, the existence of a sum-of-squares decomposition implies nonnegativity everywhere. The converse is not true without restrictions — for example the following [25] is everywhere strictly positive (geometric mean $\leq$ arithmetic mean, applied to x^2y^4, x^4y^2 and 1), but one can show by quite elementary considerations that it is not a sum of squares in $\mathbb{R}[x, y]$.

$$1 + x^4y^2 + x^2y^4 - 3x^2y^2$$

On the other hand, the positive solution of Hilbert's 17th problem [3] implies that every PSD polynomial is the sum of squares of *rational* functions. For instance, we have:

$$\begin{aligned} &1 + x^4y^2 + x^2y^4 - 3x^2y^2 = \\ &\left(\tfrac{x^2y(x^2+y^2-2)}{x^2+y^2}\right)^2 + \left(\tfrac{xy^2(x^2+y^2-2)}{x^2+y^2}\right)^2 + \left(\tfrac{xy(x^2+y^2-2)}{x^2+y^2}\right)^2 + \left(\tfrac{x^2-y^2}{x^2+y^2}\right)^2 \end{aligned}$$

We will consider in what follows a liberal notion of 'sum of squares' allowing $\sum a_i s_i(\overline{x})^2$ where the a_i are nonnegative rational numbers. This amounts to no real increase in generality since every nonnegative rational can be written as a sum of four rational squares [34]. And the reasoning that $SOS \Rightarrow PSD$ is almost equally straightforward.

Direct proof of PSD from SOS

At its simplest, we might seek to prove that a single polynomial is PSD by finding a SOS decomposition. We have seen that this approach is in general not complete. Nevertheless, in practice it often works for problems of interest, e.g. the following [12]:

$$\begin{array}{c}\forall w\ x\ y\ z.\ w^6 + 2z^2w^3 + x^4 + y^4 + z^4 + 2x^2w + 2x^2z + \\ 3x^2 + w^2 + 2zw + z^2 + 2z + 2w + 1 \geq 0\end{array}$$

via the sum-of-squares decomposition:

$$\begin{array}{l}w^6 + 2z^2w^3 + x^4 + y^4 + z^4 + 2x^2w + 2x^2z + \\ 3x^2 + w^2 + 2zw + z^2 + 2z + 2w + 1 = \\ (y^2)^2 + (x^2 + w + z + 1)^2 + x^2 + (w^3 + z^2)^2\end{array}$$

Besides its theoretical incompleteness, finding a direct SOS expansion only works for nonnegativity of a single polynomial. However, this can be generalized somewhat by a change of variables. For example, instead of proving $\forall x.\ x \geq 0 \Rightarrow p(x) \geq 0$ we can prove the equivalent $\forall x.\ p(x^2) \geq 0$. More interesting is certifying nonnegativity over a general compact interval $[a, b]$:

$$\forall x.\ a \leq x \wedge x \leq b \Rightarrow p(x) \geq 0$$

We can likewise prove this equivalent to a simple nonnegativity assertion with a change of variable. Note first that

$$x(y) = \frac{a + by^2}{1 + y^2}$$

is a surjection from $\mathbb{R}$ to $[a, b)$ with right inverse

$$y(x) = \sqrt{\frac{x - a}{b - x}}$$

and so

$$(\forall x.\ a \leq x \wedge x < b \Rightarrow p(x) \geq 0) \Leftrightarrow (\forall y \in \mathbb{R}.\ p(\frac{a + by^2}{1 + y^2}) \geq 0)$$

Moreover, because polynomials are continuous, this is equivalent to the original claim $\forall x.\ a \leq x \wedge x \leq b \Rightarrow p(x) \geq 0$. We can turn the rational function claim into purely polynomial nonnegativity by multiplying through by $(1 + y^2)^{\partial(p)}$ where $\partial(p)$ is the degree of p, since this is guaranteed to cancel all denominators:

$$(\forall x.\ a \leq x \wedge x \leq b \Rightarrow p(x) \geq 0) \Leftrightarrow (\forall y.\ (1 + y^2)^{\partial(p)} p(\frac{a + by^2}{1 + y^2}) \geq 0)$$

However, we will now consider a more general and theoretically complete approach to verifying universal formulas using SOS.

3 Important cases of Hilbert's theorem

A classic result due to Hilbert [16] shows that there are only a few special classes of polynomials where PSD and SOS are equivalent. We just note two of them.

Univariate polynomials

Every PSD *univariate* polynomial is a sum of just two real squares. For the proof, observe that complex roots always occur in conjugate pairs, and any real roots must have even multiplicity, otherwise the polynomial would cross the x-axis instead of just touching it. Thus, if the roots are $a_k \pm ib_k$, we can imagine writing the polynomial as:

$$\begin{aligned} p(x) &= [(x - [a_1 + ib_1])(x - [a_2 + ib_2]) \cdots (x - [a_m + ib_m])] \cdot \\ &\quad\ [(x - [a_1 - ib_1])(x - [a_2 - ib_2]) \cdots (x - [a_m - ib_m])] \\ &= (q(x) + ir(x))(q(x) - ir(x)) \\ &= q(x)^2 + r(x)^2 \end{aligned}$$

However, to expand a polynomial with *rational* coefficients as a sum of squares of *rational* polynomials, a more sophisticated proof is needed. For example Landau [21], building on a theorem of Hilbert, shows that every PSD univariate polynomial is the sum of 8 squares. This was subsequently sharpened by Pourchet [27] to show that 5 squares suffice, and indeed that 5 is the best possible in general. However, even the more constructive proofs of this and related results [5] do not seem to be very practical, and we will return later in this paper to finding such expansions in practice.

Quadratic forms

Every PSD quadratic form, in any number of variables, is a sum of squares. (A *form* is a polynomial where all monomials have the same [multi-]degree, and in a quadratic form that degree is 2. So for example x^2, wz and xy are permissible monomials in a quadratic form but not 1, x or y^5.) The proof is a straightforward elaboration of the elementary technique of "completing the square" [10]. We will as usual assume a standard representation of a quadratic form

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j$$

where $a_{ij} = a_{ji}$. We are at liberty to make this symmetry assumption, for given any representation we can always choose another symmetric one by setting $a'_{ij} = a'_{ji} = (a_{ij} + a_{ji})/2$.

Theorem 1. *Given a quadratic form $f(x_1, \ldots, x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j$ in variables $x_1, \ldots, x_n$ with the coefficients a_{ij} rational and $a_{ij} = a_{ji}$, we can construct either a decomposition:*

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{n} b_i g_i(x_1, \ldots, x_n)^2$$

where the b_i are nonnegative rational numbers and the $g_i(x_1, \ldots, x_n)$ are linear functions with rational coefficients, or particular rational numbers $u_1, \ldots, u_n$ such that $f(u_1, \ldots, u_n) < 0$.

Proof. By induction on the number of variables. If the form is zero, then it trivially has an empty SOS decomposition. Otherwise, pick the first variable x_1 (the order is unimportant), and separate the monomials into those involving x_1 and those not:

$$f(x_1, \ldots, x_n) = (a_{11}x_1^2 + \sum_{i=2}^{n} 2a_{1i}x_1x_i) + g(x_2, \ldots, x_n)$$

If $a_{11} = 0$, then there are two cases to consider. If all the a_{1i} are zero, then we effectively have a form in $n-1$ variables and so the result holds by induction; in the case of a witness of non-positive-semidefiniteness, we can assign u_1 arbitrarily. Otherwise, if any $a_{1i} \neq 0$, the form is not positive semidefinite and our witness is $u_1 = a_{ii}/2 + 1$, $u_i = -a_{1i}$ and all other $u_j = 0$; we then have $f(u_1, \ldots, u_n) = -2a_{1i}^2 < 0$ as required.

Now if $a_{11} < 0$, then again the form is not PSD, and a suitable witness is simply $u_1 = 1$ and all other $u_j = 0$, whence all monomials but $a_{11}x_1^2$ are zero and that one is negative. The more interesting case is when $a_{11} > 0$, and here we 'complete the square'. We have:

$$\begin{aligned} f(x_1, \ldots, x_n) = {} & a_{11}(x_1 + \textstyle\sum_{i=2}^{n}(a_{1i}/a_{11})x_i)^2 + \\ & (g(x_2, \ldots, x_n) - \textstyle\sum_{j=2}^{n}\sum_{k=2}^{n}(a_{1j}a_{1k}/a_{11})x_jx_k) \end{aligned}$$

The second term on the right is a quadratic form in variables $x_2, \ldots, x_n$, so by the inductive hypothesis we can either find a SOS expansion or a witness of non-positive-definiteness. In the former case, we just include $a_{11}(x_1 + \sum_{i=2}^{n}(a_{1i}/a_{11})x_i)^2$ and obtain a SOS decomposition for the entire form. In the latter case, we take the witness $u_2, \ldots, u_n$ and augment it by choosing $u_1 = -\sum_{i=2}^{n}(a_{1i}/a_{11})u_i$, which makes the term $a_{11}(x_1 + \sum_{i=2}^{n}(a_{1i}/a_{11})x_i)^2$ vanish and hence gives a non-PSD witness for the whole form. QED

For example, let us apply the method to the form $6x^2+49y^2+51z^2-82yz+20zx-4xy$, with the variables in the obvious order x, y, z. We obtain a SOS decomposition as follows:

$$\begin{aligned} & 6x^2 + 49y^2 + 51z^2 - 82yz + 20zx - 4xy \\ = {} & 6\left(x^2 - \frac{2}{3}xy + \frac{10}{3}xz\right) + \left(49y^2 + 51z^2 - 82yz\right) \\ = {} & 6\left(x - \frac{1}{3}y + \frac{5}{3}z\right)^2 + \left(49y^2 + 51z^2 - 82yz\right) - 6\left(-\frac{1}{3}y + \frac{5}{3}z\right)^2 \\ = {} & 6\left(x - \frac{1}{3}y + \frac{5}{3}z\right)^2 + \left(\frac{145}{3}y^2 - \frac{226}{3}yz + \frac{78}{3}z^2\right) \end{aligned}$$

$$
\begin{aligned}
&= 6\left(x - \frac{1}{3}y + \frac{5}{3}z\right)^2 + \frac{145}{3}\left(y^2 - \frac{226}{145}yz\right) + \frac{103}{3}z^2 \\
&= 6\left(x - \frac{1}{3}y + \frac{5}{3}z\right)^2 + \frac{145}{3}\left(y - \frac{113}{145}z\right)^2 + \frac{103}{3}z^2 - \frac{12769}{435}z^2 \\
&= 6\left(x - \frac{1}{3}y + \frac{5}{3}z\right)^2 + \frac{145}{3}\left(y - \frac{113}{145}z\right)^2 + \frac{722}{145}z^2
\end{aligned}
$$

4 Quadratic forms and matrices

We can establish a correspondence between quadratic forms and matrices by writing a quadratic form in variables $x_1, \ldots, x_n$ as a vector-matrix-vector product with a vector of variables:

$$\overline{x}^T A \overline{x} = \sum_{i=1}^{n} x_i \sum_{j=1}^{n} A_{ij} x_j = \sum_{1 \leq i,j \leq n} A_{ij} x_i x_j$$

If we restrict ourselves to symmetric matrices A, then the matrix representation is unique, and the matrix elements correspond exactly to the coefficients in the standard formulation above. (In an actual implementation we may choose to use an appropriately modified upper or lower triangular matrix for efficiency reasons.)

Positive semidefinite matrices

Quite generally, a symmetric[2] matrix A is said to be *positive semidefinite* iff $\overline{x}^T A \overline{x} \geq 0$ for all vectors $\overline{x}$ — in other words, precisely if the associated quadratic form is positive semidefinite. Two other equivalent characterizations are:

- There is a factorization $A = L^T L$ where L is a triangular matrix and the T signifies transposition.
- All eigenvalues of A are non-negative (they are necessarily real because A is symmetric)

A proof of the former is straightforward by recasting the “completing the square” algorithm (theorem 1) in matrix terms [31]. More precisely, a factorization of the form $A = L^T L$ is obtained by Choleski decomposition. A direct translation of the ‘completing the square’ algorithm gives a decomposition $A = LDL^T$ where L, D and L^T are respectively lower-triangular, diagonal and upper-triangular. This has some advantages for symbolic applications because it involves only rational operations, whereas Choleski decomposition requires square roots.

[2] Or Hermitian if we consider the complex case, which we will not do here.

5 The universal fragment via SOS

The Positivstellensatz

The Artin-Schreier theorem [19] implies that every ordered integral domain can be embedded in a real-closed field called its real closure. For this reason, a *universal* formula must hold in all real-closed fields (and hence, by Tarski's completeness result, in $\mathbb{R}$) iff it holds in all ordered integral domains:

- If it holds in all ordered integral domains, it holds in all real-closed fields, since a real-closed field is a kind of ordered integral domain.
- If it holds in all real-closed fields, it holds in the real closure of any ordered integral domain, and therefore, since quantifiers are all universal, in the substructure corresponding to that integral domain.

This already means that a valid formula can in principle be proved without using the special axioms about square roots and roots of odd-degree polynomials, based only on the axioms for an ordered integral domain. This can be put in a still sharper form. First it is instructive to look at the case of the complex numbers, where the classic *Hilbert Nullstellesatz* holds:

Theorem 2. *The polynomial equations $p_1(\overline{x}) = 0$, …, $p_n(\overline{x}) = 0$ have no common solution in $\mathbb{C}$ iff 1 is in the ideal generated by $p_1, \ldots, p_n$, which we write as $1 \in \mathit{Id}\,\langle p_1, \ldots, p_n\rangle$.*

More explicitly, this means that the universally quantified formula $\forall \overline{x}.\ p_1(\overline{x}) = 0 \wedge \cdots p_n(\overline{x}) = 0 \Rightarrow \bot$ holds iff there are 'cofactor' polynomials $q_1(\overline{x})$, …, $q_n(\overline{x})$ such that the following is a polynomial identity:

$$p_1(\overline{x}) \cdot q_1(\overline{x}) + \cdots + p_n(\overline{x}) \cdot q_n(\overline{x}) = 1$$

The analogous property fails over $\mathbb{R}$; for example $x^2 + 1 = 0$ alone has no solution yet 1 is not a multiple of $x^2 + 1$ (considering them as polynomials). However, in the analogous Real Nullstellensatz, sums of squares play a central role:

Theorem 3. *The polynomial equations $p_1(\overline{x}) = 0$, …, $p_n(\overline{x}) = 0$ have no common solution in $\mathbb{R}$ iff there are polynomials $s_1(\overline{x})$, …, $s_k(\overline{x})$ such that $s_1(\overline{x})^2 + \cdots + s_k(\overline{x})^2 + 1 \in \mathit{Id}\,\langle p_1, \ldots, p_n\rangle$.*

This can be further generalized to so-called 'Positivstellensatz' results on the inconsistency of a set of equations, inequations and inequalities.[3] Unfortunately these become a bit more intricate to state. The particular version we rely on in our implementation, following [26], can be stated as follows:

[3] In principle the simple Nullstellensatz suffices to prove unsatisfiability of any unsatisfiable conjunction of atomic formulas, since in $\mathbb{R}$ we have equivalences such as $s \leq t \Leftrightarrow \exists x.\ t = s + x^2$, $s < t \Leftrightarrow \exists x.\ (t - s)x^2 = 1$ and $s \neq t \Leftrightarrow \exists x.\ (t - s)x = 1$. Indeed we could then combine a conjunction of equations into a single equation using $s = 0 \wedge t = 0 \Leftrightarrow s^2 + t^2 = 0$. However, using a general Positivstellensatz tends to be more efficient.

Theorem 4. *The polynomial formulas $p_1(\overline{x}) = 0, \ldots, p_n(\overline{x}) = 0$, $q_1(\overline{x}) \geq 0, \ldots, q_m(\overline{x}) \geq 0$, $r_1(\overline{x}) \neq 0, \ldots, r_p(\overline{x}) \neq 0$ are impossible in $\mathbb{R}$ iff there are polynomials P, Q, R such that $P+Q+R^2 = 0$ where $P \in Id\ \langle p_1, \ldots, p_n \rangle$, R is a product of powers of the r_i (we can if desired assume it's of the form $\prod_{i=1}^{p} r_i^k$) and Q is in the* cone *generated by the q_i, i.e. the smallest set of polynomials containing all q_i, all squares of arbitrary polynomials and closed under addition and multiplication.*

It's perhaps easier to grasp a simple example. Consider proving the universal half of the quadratic root criterion

$$\forall a\ b\ c\ x.\ ax^2 + bx + c = 0 \Rightarrow b^2 - 4ac \geq 0$$

by showing the inconsistency of the formulas $ax^2 + bx + c = 0$ and $4ac - b^2 > 0$. We have the following polynomial identity of the form whose existence is guaranteed by the Positivstellensatz:

$$(4ac - b^2) + (2ax + b)^2 + (-4a)(ax^2 + bx + c) = 0$$

Given such a "certificate" (i.e. the additional polynomials necessary in such an equation), it's easy to verify the required result by elementary inequality reasoning: $(2ax + b)^2$ is a square and hence nonnegative, $(-4a)(ax^2 + bx + c)$ is zero since by hypothesis $ax^2 + bx + c = 0$, so $b^2 - 4ac$ must be nonnegative for the equation to hold.

Finding SOS decompositions by semidefinite programming

Although the Nullstellensatz/Positivstellensatz assures us that suitable SOS certificates of infeasibility exist, the usual proofs of these results are not constructive. Lombardi [22] has proved a constructive form, showing how a refutation using Hörmander's procedure can be used to systematically construct a Nullstellensatz certificate. However, given that we know that none of the sophisticated real-closed field axioms are actually needed, we might seek a more direct approach.

Parrilo [26] pioneered the approach of using *semidefinite programming* to find SOS decompositions. Semidefinite programming is the problem of finding feasible values $u_1, \ldots, u_m$ (or more generally, maximizing some linear combination of the u_i) to make a matrix linearly parametrized by those values PSD, subject to a set of linear equational constraints on the u_i. This is a convex optimization problem, and so in principle we know it can be solved to a given accuracy in polynomial time, e.g. on the basis of the ellipsoid algorithm [2, 11].[4] In practice, there are powerful semidefinite programming tools, e.g. based on primal-dual interior point algorithms or the 'spectral bundle method'. Our experiments have mostly used the system CSDP [6], which we have found to be robust and efficient.[5]

[4] As [29] notes, convexity rather than linearity is the fundamental property that makes optimization relatively tractable. Indeed, the first polynomial-time algorithm for linear programming [20] was based on the ellipsoid algorithm for general convex optimization, together with an argument about about the accuracy bound needed.

[5] See also the CSDP Web page `https://projects.coin-or.org/Csdp/`.

The basic idea of this reduction is to introduce new variables for the possible monomials that could appear in the squares. For example [26] to express $2x^4+2x^3y-x^2y^2+5y^4$ as a SOS, no monomials of degree > 2 can appear in the squares, since their squares would then remain uncancelled in the SOS form. With a little more care one can deduce that only the following monomials, for which we introduce the new variables z_i, need be considered:

$$z_1 = x^2,\ z_2 = y^2,\ z_3 = xy$$

Now we write the original polynomial as a quadratic form in the z_i:

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}^T \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

Comparing coefficients in the original coefficient, we obtain linear constraints on the q_{ij}:

$$\begin{aligned} &q_{11} = 2 \\ &q_{22} = 5 \\ &q_{33} + 2q_{12} = -1 \\ &2q_{13} = 2 \\ &2q_{23} = 0 \end{aligned}$$

By introducing the new variables, we have returned to the case of quadratic forms, where SOS and PSD are the same. Thus, if we find q_{ij} for which the matrix is PSD, we can directly read off a sum of squares using the ‘completing the square’ algorithm. The price we pay is that only solutions satisfying the linear constraints above will yield an SOS expansion for the original polynomial; for our example $q_{33} = 5$ and $q_{12} = -3$ give such a solution, from which a SOS decomposition can be read off. SDP solvers can solve exactly this problem.

When just searching for a direct SOS decomposition as in the example above, we were able to place a bound on the monomials we need to consider. However, for general Positivstellensatz certificates, the only bounds known are somewhat impractical. Instead of using these, we impose relatively small limits on the degrees of the polynomials considered (in the squares and in the ideal cofactors) as well as the powers used for the product of inequations. For any particular bound, the problem reduces to semidefinite programming, and we can keep increasing the bound until it succeeds, at least in principle.

6 Implementation in HOL and experience

We have integrated this algorithm into HOL Light, and it is freely available in the latest version (2.20) as the file `Examples/sos.ml`.

How it works

We rely on all the existing machinery in HOL Light for eliminating various auxiliary concepts like the absolute value function, $\max$ and $\min$, and reducing the problem to a finite set of subproblems that involve just refuting a conjunction of equations, strict and nonstrict inequalities. All this is currently used in the simple linear prover of HOL Light, and it was designed so that any other core equation/inequality refuter can be plugged in. So now the task is just to refute a conjunction of the form:

$$\bigwedge_i p_i(\overline{x}) = 0 \wedge \bigwedge_j q_j(\overline{x}) \geq 0 \wedge \bigwedge_k r_k(\overline{x}) > 0$$

We do this by systematically searching for certificates of the form guaranteed to exist by the Positivstellensatz. We use iterative deepening, so that at stage n we consider, roughly speaking, polynomials of degree n in the certificate. Symbolically, using the underlying programming language OCaml in which HOL Light is implemented, we invent parametrized polynomials for the certificate and solve the constraints that result from comparing coefficients, to obtain a semidefinite programming problem. The semidefinite programming problem is printed to a file and we call a semidefinite programming package to solve it.

Then comes a rather tricky aspect. The vector returned by the SDP solver is of floating-point numbers, and we need to translate them to rational numbers in HOL. Unfortunately if we map them to the exact rational they denote, we very often find that the resulting matrix is *not* quite PSD, because of the influence of rounding errors (e.g. 0.33333333333 instead of exactly $1/3$). So instead we postulate that the “exact” solutions probably involve rational numbers with moderate coefficients, and try rounding all the values based on common denominators $1, 2, 3, \ldots$. (Once we reach 32 we start to go up by a multiple of 2 each time.) For each attempt we test, using exact rational arithmetic in OCaml, whether the resulting matrix is PSD, using the algorithm from theorem 1. As soon as we find a case where it is, we extract the SOS decomposition and prove the resulting identity in HOL.

Generally speaking this seems to work quite well. It is interesting to note that although we only seek feasible solutions, we tend to do better when rounding if we try to optimize something (arbitrarily, we minimize the sum of the diagonal elements of the matrix). So it seems that extremal solutions tend to involve nicer rational numbers than arbitrary points. We originally attempted to find good ‘simple’ rational approximants to the coefficients independently, but the approach of picking a single common denominator seems to work better.

Examples

The primary interface is `REAL_SOS`, which attempts to prove a purely universal formula over the reals. Here is a typical user invocation:

```
# REAL_SOS
  `a1 >= &0 /\ a2 >= &0 /\
   (a1 * a1 + a2 * a2 = b1 * b1 + b2 * b2 + &2) /\
   (a1 * b1 + a2 * b2 = &0)
   ==> a1 * a2 - b1 * b2 >= &0`;;
```

and the output from HOL Light:

```
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Translating proof certificate to HOL
val it : thm =
  |- a1 >= &0 /\
     a2 >= &0 /\
     a1 * a1 + a2 * a2 = b1 * b1 + b2 * b2 + &2 /\
     a1 * b1 + a2 * b2 = &0
     ==> a1 * a2 - b1 * b2 >= &0
```

The informative messages indicate the iterative deepening of the bounds imposed; at each stage the semidefinite programming problem is passed to CSDP for solution. At a bound of 4 the underlying semidefinite programming problem involves a 38×38 matrix (in 8 block diagonal portions, which SDP solvers can exploit) parametrized by 143 variables. This problem is solved and HOL Light succeeds in rounding the output vector to a certificate and hence proving the original claim. The entire sequence of events takes around one second. Numerous similar instances of classic elementary inequalities can be proved efficiently, including almost all the examples in [12] and the following:

```
# REAL_SOS
   `&0 <= x /\ &0 <= y
    ==> x * y * (x + y) pow 2 <= (x pow 2 + y pow 2) pow 2`;;
```

On top of `REAL_SOS`, we have also implemented analogous functions for $\mathbb{Z}$ and $\mathbb{N}$, which do some elimination of division and modulus followed by simple-minded and incomplete discretization (e.g. translating hypotheses of the form $x < y$ to $x \leq y - 1$) then call the real version. This is enough to solve a few not entirely trivial properties of truncating division on $\mathbb{N}$, e.g.

```
# SOS_RULE
  `!a b c d. ~(b = 0) /\ b * c < (a + 1) * d ==> c DIV d <= a DIV b`;;
```

and

```
# SOS_RULE `0 < m /\ m < n  ==> ((m * ((n * x) DIV m + 1)) DIV n = x)`;;
```

Problems

This approach to verifying nonlinear inequalities, and more generally to nonlinear optimization, has much to recommend it on general grounds [26]. But in the context of a foundational theorem prover it is especially appealing, because the “difficult” part, finding the certificates, can be done using highly optimized, unverified external programs. The theorem prover merely needs to *verify* the certificate [15]. We have found it fast and powerful enough to prove many lemmas that come up in practice as part of larger proofs. Quite recently we were happy to use it for the otherwise slightly tedious lemmas $0 \leq a \wedge 0 \leq b \wedge 0 \leq c \wedge c(2a + b)^3/27 \leq x \Rightarrow ca^2b \leq x$ (1.25 seconds) and $-1 \leq t \wedge t \leq 1 \Rightarrow 0 \leq 1 + r^2 - 2rt$ (0.06 seconds). However, we have encountered two persistent problems that suggest necessary improvements.

First, sometimes our naive rounding procedure is not adequate, and even though the SDP solver seems to solve the semidefinite program fairly accurately, none of the roundings we try results in a PSD matrix. In this case, we end up in an infinite loop, exploring more complex certificates without any benefit. It would certainly be desirable to have a more intelligent approach to rounding, but at present we are not sure what the best approach is. Indeed, in principle the exact floating-point result can be platform-dependent, since the involved numerical algorithms underlying SDP have often been optimized in slightly different ways (e.g. a different order of operations when multiplying matrices).

Second, the treatment of strict inequalities in our Positivstellensatz (considering $p > 0$ as $p \geq 0$ and $p \neq 0$) means that the style of exploration of the search space depends critically on whether the inequalities are strict or non-strict. This can sometimes have tiresome consequences. For example the following example (which I got from Russell O'Connor) works quite quickly:

```
# REAL_SOS
  `a * a + a * b - b * b >= &0 /\
   &2 * a + b >= &0 /\
   c * c + c * d - d * d <= &0 /\
   d >= &0
   ==> a * d + c * b + b * d >= &0`;;
```

If we replace a few non-strict ($\geq$) inequalities in the hypotheses by strict ones ($>$), we might expect it to become *easier*. Yet because of the difference in treatment of strict inequalities, the problem is only solved at a much higher depth, and moreover at that point the rounding problem has appeared and means that we do not solve it at all!

7 Optimizing the univariate case

Most floating-point transcendental function implementations ultimately rely on a polynomial approximation over an interval, and we would like to verify an error bound theorem of the form $\forall x. a \leq x \leq b \Rightarrow |f(x) - p(x)| \leq \epsilon$ relating the function f to its polynomial approximation. By choosing a very accurate Taylor series expansion $t(x)$, one can reduce the problem to bounding a polynomial $\forall x.\ a \leq x \leq b \Rightarrow |t(x) - p(x)| \leq \epsilon$. This is a problem that has sometimes preoccupied the present author for some time, and formally verified solutions can be quite lengthy to compute [13, 14]. The idea of proving such bounds using SOS techniques, even a direct SOS decomposition after change of variables, is very attractive.

Unfortunately, the numerical difficulties mentioned above are a serious issue here, and we have not had much success with SOS methods except on artificially simple examples. It is not hard to understand why these cases are numerically difficult. The coefficients of $p(x)$ are carefully chosen to minimize the maximum error over the interval, and are not simple rational numbers. Moreover, by design, the error bounds tend to be small relative to the coefficients. (A simple and idealized form of the same phenomenon is that the Chebyshev polynomials $T_n(x)$ are bounded by 1 over the interval $[-1, 1]$ even though their leading coefficient is 2^n.)

We therefore consider a different approach, where we adapt the simple proof that every univariate PSD polynomial is a sum of two real squares to find exact rational decompositions. This has not yet been completely automated and integrated into HOL Light, but we have a simple script that runs in PARI/GP[6] and appears to be promising. (We rely on the excellent arbitrary-precision complex root finder in PARI/GP, which implements a variant of an algorithm due to Schönhage.) We will explain the algorithm in general, as well as tracing a specific example:

$$\begin{aligned} p(x) &= ((x-1)^8 + 2(x-2)^8 + (x-3)^8 - 2)/4 \\ &= x^8 - 16x^7 + 126x^6 - 616x^5 + 1995x^4 - 4312x^3 + 6006x^2 - 4888x + 1768 \end{aligned}$$

Elimination of repeated roots

If a polynomial is ≥ 0 everywhere, then real roots must occur with even multiplicity. Therefore, we start out as in squarefree decomposition by taking $d = \gcd(p, p')$ and writing $p = d^2 q$. The remaining polynomial q must have no real roots, and hence be strictly > 0 everywhere. If we can write q as a SOS, we can just multiply inside each square by d and get a SOS for p. In our running example, this factors out one repeated root $x = 2$:

$$\begin{aligned} p &= x^8 - 16x^7 + 126x^6 - 616x^5 + 1995x^4 - 4312x^3 + 6006x^2 - 4888x + 1768 \\ p' &= 8x^7 - 112x^6 + 756x^5 - 3080x^4 + 7980x^3 - 12936x^2 + 12012x - 4888 \\ d &= x - 2 \\ q &= x^6 - 12x^5 + 74x^4 - 272x^3 + 611x^2 - 780x + 442 \end{aligned}$$

Note that this step requires only rational operations and does not introduce any inaccuracy.

Perturbation

Since all polynomials of odd degree have a real root, the degree of the original p and our polynomial q must be even, say $\partial(q) = n = 2m$. Since it is *strictly* positive, there must be an $\epsilon > 0$ such that the perturbed polynomial

$$q - \epsilon(1 + x^2 + ... + x^{2m})$$

is also (strictly) positive. To find such an ϵ we just need to test if a polynomial has real roots, which we can easily do in PARI/GP; we can then search for a suitable ϵ by choosing a convenient starting value and repeatedly dividing by 2 until our goal is reached; we actually divide by 2 again to leave a little margin of safety. (Of course, there are more efficient ways of doing this.) In this case we get $\epsilon = 1/32$ and the perturbed polynomial becomes:

$$31/32x^6 - 12x^5 + 2367/32x^4 - 272x^3 + 19551/32x^2 - 780x + 14143/32$$

We have been assuming that the initial polynomial *is* indeed PSD, but if it is not, that fact will be detected at this stage by checking the $\epsilon = 0$ case.

[6] `http://pari.math.u-bordeaux.fr/`

Approximate SOS of perturbed polynomial

We now use the basic 'sum of two real squares' idea to obtain an approximate SOS decomposition of the perturbed polynomial r, just by using approximations of the roots, close enough to make the final step below work correctly. Now we have $r = ls^2 + lt^2 + u$ where l is the leading coefficient of r, such that the remainder u is relatively small. Using our PARI/GP script on the running example we obtain for this remainder:

$$\begin{aligned} u = {} & 7/65536x^5 - 522851/268435456x^4 + 1527705/268435456x^3 - \\ & 655717/536870912x^2 - 14239/2097152x + 1913153/536870912 \end{aligned}$$

Absorption of remainder term

We now have $q = ls^2 + lt^2 + \epsilon(1 + x^2 + ... + x^{2m}) + u$, so it will suffice to express $\epsilon(1 + x^2 + ... + x^{2m}) + u$ as a sum of squares. Note that the degree of u is $< 2m$ by construction (though the procedure to be outlined would work with minor variations even if it were exactly $2m$). Let us say $u = a_0 + a_1 x + \ldots + a_{2m-1}x^{2m-1}$. Note that

$$\begin{aligned} x &= (x + 1/2)^2 - (x^2 + 1/4) \\ -x &= (x - 1/2)^2 - (x^2 + 1/4) \end{aligned}$$

and so for any $c \geq 0$:

$$\begin{aligned} cx^{2k+1} &= c(x^{k+1} + 1/2x^k)^2 - c(x^{2k+2} + 1/4x^{2k}) \\ -cx^{2k+1} &= c(x^{k+1} - 1/2x^k)^2 - c(x^{2k+2} + 1/4x^{2k}) \end{aligned}$$

Consequently we can rewrite the odd-degree terms of u as

$$a_{2k+1}x^{2k+1} = |a_{2k+1}|(x^{k+1} + \text{sgn}(a_{2k+1})/2x^k)^2 - |a_{2k+1}|(x^{2k+2} + 1/4x^{2k})$$

and so:

$$\begin{aligned} \epsilon(1 + x^2 + ... + x^{2m}) + u = {} & \textstyle\sum_{k=0}^{m-1} |a_{2k+1}|(x^{k+1} + \text{sgn}(a_{2k+1})/2x^k)^2 + \\ & \textstyle\sum_{k=0}^{m} (\epsilon + a_{2k} - |a_{2k-1}| - |a_{2k+1}|/4)x^{2k} \end{aligned}$$

where by convention $a_{-1} = a_{2m+1} = 0$. This already gives us the required SOS representation, provided each $\epsilon + a_{2k} - |a_{2k-1}| - |a_{2k+1}|/4 \geq 0$, and we can ensure this by computing the approximate SOS sufficiently accurately. In the running example, our overall expression is

$$\begin{aligned} & 31/32(x^3 - 25369/4096x^2 + 313/64x + 32207/4096)^2 + \\ & 31/32(21757/4096x^2 - 90963/4096x + 1271/64)^2 + \\ & 14239/2097152(x - 1/2)^2 + \\ & 1527705/268435456(x^2 + 1/2x)^2 + \\ & 7/65536(x^3 + 1/2x^2)^2 + \\ & 2041/65536x^6 + \\ & 1582721/67108864x^4 + \\ & 23424925/1073741824x^2 + \\ & 17779073/536870912 \end{aligned}$$

and we can recover a SOS decomposition for the original polynomial by incorporating the additional factor $x - 2$ into each square:

$$\begin{array}{l}
31/32(x^4 - 33561/4096x^3 + 35385/2048x^2 - 7857/4096x - 32207/2048)^2 + \\
31/32(21757/4096x^3 - 134477/4096x^2 + 131635/2048x - 1271/32)^2 + \\
14239/2097152(x^2 - 5/2x + 1)^2 + \\
1527705/268435456(x^3 - 3/2x^2 - x)^2 + \\
7/65536(x^4 - 3/2x^3 - x^2)^2 + \\
2041/65536(x^4 - 2x^3)^2 + \\
1582721/67108864(x^3 - 2x^2)^2 + \\
23424925/1073741824(x^2 - 2x)^2 + \\
17779073/536870912(x - 2)^2
\end{array}$$

8 Conclusions and related work

Our work so far shows that SOS is a very promising approach to this class of problem. Despite the implementation in a very foundational theorem-prover, simple problems are solved fast enough to be a real boon in practice. However, we have also noted a couple of difficulties. The rounding problem seems to be the most pressing. It could be avoided given an arbitrary-precision package for semidefinite programming. However, replacing ordinary floating-point arithmetic in a semidefinite programming engine with something like MPFR[7] would be highly non-trivial even for their designers, since they are complex programs depending on an infrastructure of linear algebra packages. Lacking a high-precision SDP solver, we need to come up with a more ‘intelligent’ approach to rounding, but this seems non-trivial. As an answer to our problems with strict inequalities, there are numerous possibilities, such as directly eliminating strict inequalities in terms of equations or using a different form of Positivstellensatz, even a radically different one such as Schmüdgen’s [28]. We have already experimented with other valuable optimizations, e.g. exploiting symmetry and using Gröbner bases to handle equations, and these should be pursued and properly integrated into the mainstream version.

Acknowledgements

I want to thank Henri Lombardi for first taking the time to explain some of the basic theory of the Real Nullstellensatz to me, so setting me off on a long voyage of discovery. The debt to Pablo Parrilo is clear: a large part of this work is just an implementation of his key ideas. The ideas for optimizing the univariate case and its implementation in PARI/GP were worked out when I was a visitor in the Arenaire group at the ENS in Lyon. I am grateful to many people there for stimulating discussions, and in particular to Christoph Lauter for help with German translation and for some additional examples. I also benefited from the discussion at CMU when I presented SOS at a seminar organized by Jeremy Avigad and Ed Clarke. Laurent Théry, besides porting some of the work to Coq, persuaded me that I should write things up for publication. I’m also grateful to the anonymous referees for their constructive suggestions and for catching several errors.

[7] `http://www.mpfr.org/`

References

1. B. Akbarpour and L. C. Paulson. Towards automatic proofs of inequalities involving elementary functions. In B. Cook and R. Sebastiani, editors, *Proceedings of PDPAR 2006: Pragmatics of Decision Procedures in Automated Reasoning*, pages 27–37, 2006.
2. M. Akgül. *Topics in relaxation and ellipsoidal methods*, volume 97 of *Research notes in mathematics*. Pitman, 1984.
3. E. Artin. Über die Zerlegung definiter Funktionen in Quadrate. *Hamburg Abhandlung*, 5:100–115, 1927.
4. J. Avigad and H. Friedman. Combining decision procedures for the reals. To appear in "Logical Methods in Computer Science". Available online at `http://arxiv.org/abs/cs.LO/0601134`., 2006.
5. S. Basu. A constructive algorithm for 2-D spectral factorization with rational spectral factors. *IEEE Transactions on Circuits and Systems–I: Fundamental Theory and Applications*, 47:1309–1318, 2000.
6. B. Borchers. CSDP: A C library for semidefinite programming. *Optimization Methods and Software*, 11:613–623, 1999.
7. B. F. Caviness and J. R. Johnson, editors. *Quantifier Elimination and Cylindrical Algebraic Decomposition*, Texts and monographs in symbolic computation. Springer-Verlag, 1998.
8. P. J. Cohen. Decision procedures for real and p-adic fields. *Communications in Pure and Applied Mathematics*, 22:131–151, 1969.
9. G. E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In H. Brakhage, editor, *Second GI Conference on Automata Theory and Formal Languages*, volume 33 of *Lecture Notes in Computer Science*, pages 134–183, Kaiserslautern, 1976. Springer-Verlag.
10. W. L. Ferrar. *Algebra: a text-book of determinants, matrices, and algebraic forms*. Oxford University Press, 2nd edition, 1957.
11. M. Grotschel, L. Lovsz, and A. Schrijver. *Geometric algorithms and combinatorial optimization*. Springer-Verlag, 1993.
12. Z. Guangxing and Z. Xiaoning. An effective decision method for semidefinite polynomials. *Journal of Symbolic Computation*, 37:83–99, 2004.
13. J. Harrison. Verifying the accuracy of polynomial approximations in HOL. In E. L. Gunter and A. Felty, editors, *Theorem Proving in Higher Order Logics: 10th International Conference, TPHOLs'97*, volume 1275 of *Lecture Notes in Computer Science*, pages 137–152, Murray Hill, NJ, 1997. Springer-Verlag.
14. J. Harrison. Formal verification of floating point trigonometric functions. In W. A. Hunt and S. D. Johnson, editors, *Formal Methods in Computer-Aided Design: Third International Conference FMCAD 2000*, volume 1954 of *Lecture Notes in Computer Science*, pages 217–233. Springer-Verlag, 2000.
15. J. Harrison and L. Théry. A sceptic's approach to combining HOL and Maple. *Journal of Automated Reasoning*, 21:279–294, 1998.
16. D. Hilbert. Über die Darstellung definiter Formen als Summe von Formenquadraten. *Mathematische Annalen*, 32:342–350, 1888.
17. L. Hörmander. *The Analysis of Linear Partial Differential Operators II*, volume 257 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, 1983.
18. W. A. Hunt, R. B. Krug, and J. Moore. Linear and nonlinear arithmetic in ACL2. In D. Geist, editor, *Proceedings of the 12th Advanced Research Working Conference on Correct Hardware Design and Verification Methods, CHARME 2003*, volume 2860 of *Lecture Notes in Computer Science*, pages 319–333. Springer-Verlag, 2003.
19. N. Jacobson. *Basic Algebra II*. W. H. Freeman, 2nd edition, 1989.

20. L. G. Khachian. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979.
21. E. Landau. Über die Darstellung definiter Funktionen durch Quadrate. *Mathematischen Annalen*, 62:272–285, 1906.
22. H. Lombardi. Effective real nullstellensatz and variants. In T. Mora and C. Traverso, editors, *Proceedings of the MEGA-90 Symposium on Effective Methods in Algebraic Geometry*, volume 94 of *Progress in Mathematics*, pages 263–288, Castiglioncello, Livorno, Italy, 1990. Birkhäuser.
23. A. Mahboubi and L. Pottier. Elimination des quantificateurs sur les réels en Coq. In Journées Francophones des Langages Applicatifs (JFLA), available on the Web from `http://pauillac.inria.fr/jfla/2002/actes/index.html08-mahboubi.ps`, 2002.
24. S. McLaughlin and J. Harrison. A proof-producing decision procedure for real arithmetic. In R. Nieuwenhuis, editor, *CADE-20: 20th International Conference on Automated Deduction, proceedings*, volume 3632 of *Lecture Notes in Computer Science*, pages 295–314, Tallinn, Estonia, 2005. Springer-Verlag.
25. T. S. Motzkin. The arithmetic-geometric inequality. In O. Shisha, editor, *Inequalities*. Academic Press, 1967.
26. P. A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, 96:293–320, 2003.
27. Y. Pourchet. Sur la répresentation en somme de carrés des polynômes a une indeterminée sur un corps de nombres algébraiques. *Acta Arithmetica*, 19:89–109, 1971.
28. A. Prestel and C. N. Dalzell. *Positive Polynomials: From Hilbert's 17th Problem to Real Algebra*. Springer monographs in mathematics. Springer-Verlag, 2001.
29. R. T. Rockafellar. Lagrange multipliers and optimality. *SIAM review*, 35:183–283, 1993.
30. A. Seidenberg. A new decision method for elementary algebra. *Annals of Mathematics*, 60:365–374, 1954.
31. G. Strang. *Linear Algebra and its Applications*. Brooks/Cole, 3rd edition, 1988.
32. A. Tarski. *A Decision Method for Elementary Algebra and Geometry*. University of California Press, 1951. Previous version published as a technical report by the RAND Corporation, 1948; prepared for publication by J. C. C. McKinsey. Reprinted in [7], pp. 24–84.
33. A. Tiwari. An algebraic approach to the satisfiability of nonlinear constraints. In *Computer Science Logic, 19th International Workshop CSL 2005*, volume 3634 of *Lecture Notes in Computer Science*, pages 248–262. Springer-Verlag, 2005.
34. A. Weil. *Number Theory: An approach through history from Hammurapi to Legendre*. Birkhäuser, 1983.