SOS and SDP for the universal theory of reals

John Harrison
Intel Corporation

Decision Procedures Forum, Cambridge

Mon 12th September 2005 (09:15 – 09:45)

Summary

- The theory of reals and its universal fragment
- Nonnegativity via sum-of-squares
- Semidefinite programming
- The real Positivstellensatz
- Experiences

Disclaimer

My role here is ambassadorial. Nothing of substance in what follows is original to me.

Based heavily on work of Parrilo, and also older work by Shor etc.

General point: learn the lessons of related fields (optimization, constraints, computer algebra, . . .) as well as other theorem provers (Mizar, ACL2, . . .)

The theory of reals and its universal fragment

Consider this first-order theory:

- All rational constants p/q
- Operators of negation, addition, subtraction and multiplication
- Relations '$=$', '$<$', '$\leq$', '$>$', '$\geq$'

An interesting theory that can express many nontrivial (indeed open) problems:

> Kissing problem: how many disjoint n-dimensional spheres can be packed into space so that they touch a given unit sphere?

Decidability

There is a quantifier elimination algorithm, and so a decision method, for this theory (Tarski).

$$\mathbb{R} \models (\exists x.\, ax^2 + bx + c = 0) \Leftrightarrow a \neq 0 \wedge b^2 \geq 4ac \vee a = 0 \wedge (b \neq 0 \vee c = 0)$$

Collins's CAD algorithm is much more efficient and the first decision method actually to be implemented.

Some good implementations like `qepcad` and `REDLOG`, but theoretical and practical complexity issues limit its application.

Cohen-Hörmander algorithm is significantly simpler and has been implemented in Coq and HOL to generate proofs, but even slower.

The universal fragment

Many interesting problems fall into the purely universal fragment:

- Everyday trivialities like $\forall x\ y.\ x \geq 0 \wedge y \geq 0 \Rightarrow xy \geq 0$
- Polynomial bound problems like $\forall x \in [0, 1].\ |p(x)| \leq k$ (used for some of my verifications).
- Most classical geometrical theorems

NB: geometry theorems with no use of ordering often turn out to be true over $\mathbb{C}$, which makes things easier.

Positivity

Consider first an even more special case of proving *positive semidefiniteness*:

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

Not as limited as it may appear: can express polynomial bounds by change of variables like $x \mapsto \frac{y^2}{1+y^2}$

Illustrates the core techniques of SOS and SDP methods while avoiding some technicalities.

Sum-of-squares proofs

A *sufficient* condition for

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

is the expressibility of p as a sum of squares (SOS)

$$p(x_1, \ldots, x_n) = s_1(x_1, \ldots, x_n)^2 + \cdots + s_k(x_1, \ldots, x_n)^2$$

It is *not* a necessary condition, as shown by the *Motzkin form* $1 + x^4y^2 + x^2y^4 - 3x^2y^2$. But:

- In practice, nonnegativity problems often are solvable via SOS
- Can base complete approaches on similar SOS methods

Example (problem)

Consider the following (Zeng et al, JSC vol 37, 2004, p83-99).

$$\forall w\ x\ y\ z.\quad w^6 + 2z^2w^3 + x^4 + y^4 + z^4 + 2x^2w + 2x^2z + \\ 3x^2 + w^2 + 2zw + z^2 + 2z + 2w + 1 \geq 0$$

Constraint problems of this sort are in general quite hard to solve.

Example (solution)

We can express the polynomial as a SOS:

$$w^6 + 2z^2w^3 + x^4 + y^4 + z^4 + 2x^2w + 2x^2z +$$
$$3x^2 + w^2 + 2zw + z^2 + 2z + 2w + 1 =$$
$$(y^2)^2 + (x^2 + w + z + 1)^2 + x^2 + (w^3 + z^2)^2$$

Note how nice this is for LCF-style proving: the SOS decomposition can be checked without any tricky decision procedures.

But how do we find the SOS decomposition? By semidefinite programming (SDP)!

Reduction to quadratic form

By introducing new variables for monomials, we can express a polynomial as a quadratic form subject to linear constrants. Example:

$$2x^4 + 2x^3y - x^2y^2 + 5y^2$$

We consider all monomials (only need homogenous ones since original is a form): $z_1 = x^2,\ z_2 = y^2,\ z_3 = xy$ and write the polynomial in matrix form:

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}^T \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

Linear parametrization

By comparing coefficients we get linear constraints, and after exploiting symmetry, we only end up with one parameter.

$$q_{11} = 5$$
$$q_{22} = 5$$
$$q_{33} + 2q_{12} = -1$$
$$2q_{13} = 2$$
$$2q_{23} = 0$$

In general we'll get more, but the key point is that the parametrization is linear.

Semidefinite programming

In general, we noted that being positive semidefinite is not equivalent to having an SOS decomposition.

But for quadratic forms it *is* (basically just ‘completing the square’).

Finding a parametrization making a matrix PSD, subject to (and optimizing) linear constraints is a standard problem called *semidefinite programming*.

The problem is polynomial-time solvable using interior-point algorithms.

There are many efficient tools to solve the problem effectively in practice. I mostly use CSDP.

The usual Nullstellensatz

Over algebraically closed fields like $\mathbb{C}$ we have a nice simple equivalence.

The polynomial equations $p_1(x_1, \ldots, x_n) = 0, \ldots, p_k(x_1, \ldots, x_n) = 0$ in an algebraically closed field have *no* common solution iff there are polynomials $q_1(x_1, \ldots, x_n), \ldots, q_k(x_1, \ldots, x_n)$ such that the following polynomial identity holds:

$$q_1(x_1, \ldots, x_n) \cdot p_1(x_1, \ldots, x_n) + \cdots + q_k(x_1, \ldots, x_n) \cdot p_k(x_1, \ldots, x_n) = 1$$

Thus we can reduce equation-solving to ideal membership and solve it efficiently using Gröbner bases.

Real Nullstellensatzen and Positivstellensatzen

There is a similar but more complicated Nullstellensatz (and Positivstellensatz) over $\mathbb{R}$.

The general form is similar, but it's more complicated because of all the different orderings.

It inherently involves sums of squares (SOS), and the certificates can be found efficiently using semidefinite programming.

Example: prove

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

via the following SOS certificate:

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

Experience

This approach is often much more efficient than competing techniques such as general quantifier elimination.

Lends itself very well to a separation of proof search and LCF-style checking, so fits very well with HOL Light.

Still some awkward numerical problems where the PSD is tight (can become zero) and the rounding to rationals causes loss of PSD-ness.

Available with HOL Light 2.0 release in `Examples/sos.ml`, and seems quite useful.