On the Generation of Positivstellensatz Witnesses in Degenerate Cases

Working around geometrical degeneracy in semidefinite programming

David Monniaux

CNRS / VERIMAG
A joint laboratory of CNRS and Université Joseph Fourier (Grenoble).

August 25, 2011

Witnesses of unsatisfiability

To show that F is satisfiable: exhibit $\mathcal{M}$ such that $\mathcal{M} \models F$.
But to show that F is unsatisfiable? *Negativa non sunt probanda*.

Witness conveys confidence of proof — opposed to blind trust in a system saying "unsat".

Linear real inequalities

Farkas' lemma:

$$\begin{cases} L_1(x, y, \dots) & \leq & C_1 \\ \vdots & \vdots & \vdots \\ L_n(x, y, \dots) & \leq & C_n \end{cases}$$

has no solution iff $\exists \lambda_1, \dots, \lambda_n \geq 0$ st $\sum_i \lambda_i L_i$ is the null linear form and $\sum_i \lambda_i C_i < 0$, so the combination is $0 \leq -1$ (or any $0 \leq C$ where $C < 0$).

Looking for such λ_i= finding a solution to dual system of (in)equalities.

Complex polynomial equalities

Hilbert's Nullstellensatz (on $\mathbb{C}$):

$$\begin{cases} P_1(x, y, \dots) & = & 0 \\ \vdots & \vdots & \vdots \\ P_n(x, y, \dots) & = & 0 \end{cases}$$

has no solution iff 1 belongs to the ideal I generated by $P_1, \dots, P_n$: $\exists Q_1, \dots, Q_n$ st $\sum_i Q_i P_i = 1$.

Such Q_i can be computed by dividing 1 by a Gröbner basis for P_i (e.g. Buchberger's algorithm).

Good witnesses

Usage: convince people (or Coq or whatever) that a system is unsatisfiable by giving them a witness proving that.

An unsatisfiability witness for a system of relations should be easy to check

- low algorithm and implementation complexity (checking phase should be easy to understand)
- ideally, low time and space complexity

In both previous examples, checking the witness is simple formal math, finding it is harder.

Tackle: polynomial real inequalities.

Hilbert's 17th problem

Proving that $P < 0$ is unsatisfiable $\Leftrightarrow$ proving $P \geq 0$.

One method: prove that P is a sum of squares of polynomials (SOS).
Not a complete method: some $P \geq 0$ are not sums of squares of polynomials.

Artin: any $P \geq 0$ is a sum of squares of rational functions.
Thus: any $P \geq 0$ is N/D where N, D sums of squares of polynomials.
Equivalently: $PD - N = 0$, $D \neq 0$

Positivstellensatz

Stengle, 1973

$$\begin{cases} P_1(x, y, \dots) & \geq & 0 \\ \vdots & \vdots & \vdots \\ P_n(x, y, \dots) & \geq & 0 \end{cases}$$

has no solution iff there exists Q_j sum of squares of polynomials such that $\sum_j Q_j \tilde{P}_j = -1$, where the $\tilde{P}_j$ are the 2^n products of the form $\prod_i P_i^{\alpha_i}$ where $\alpha_i \in \{0, 1\}$.

The sums-of-squares problem

Given $P_1, \ldots, P_n, R \in \mathbb{Q}[X_1, \ldots, X_m]$, solve

$$\sum_i P_i Q_i = R$$

where the Q_i are sums of squares of polynomials in $\mathbb{Q}[X_1, \ldots, X_m]$.

Reduction to semidefinite programming

$Q \in \mathbb{Q}[X_1, \ldots, X_m]$ is a SOS polynomial over monomials $m_1, \ldots, m_l$ iff $Q = m\hat{Q}m^T$ with m vector $m_1, \ldots, m_l$ and $\hat{Q}$ is a positive semidefinite (sdp) rational matrix.

Sdp matrix = symmetric matrix with nonnegative eigenvalues, noted $\hat{Q} \succeq 0$

Thus $P_i Q_i = R$ iff there exist sdp matrices $\hat{Q}_i$ st $\sum_i P_i(m\hat{Q}_i m^T) = R$.

Equality between polynomials iff equality of coefficients for all monomials. Write this as system of equalities between the coefficients of $\hat{Q}_i$.

Reduction to semidefinite programming

The coefficients of $\hat{Q}$ satisfy a given system of linear equalities
$\Rightarrow$ solve them for a system of generators.

Then find sdp combination of the generators.

$=$ semidefinite programming feasibility problem, solved by interior point methods

Summary so far

(See e.g. Harrison TPHOL 2007)

We have reduced unsatisfiability witness search problems to:

- defining some monomial basis $m_1, \ldots, m_l$ (no good bounds on the degrees of the monomials needed, in general)
- looking for a rational solution to a pure feasibility semidefinite programming problem $-F_0 + \sum_i \lambda_i F_i \succeq 0$.

(Note: general semidefinite programming = optimize a linear form over the solution set.)

How do we solve the semidefinite programming problem? Numerically, but...

The spectrahedron

The locus of the $(\lambda_1, \ldots, \lambda_d)$ such that $-F_0 + \sum_i \lambda_i F_i \succeq 0$ is sometimes called the spectrahedron.

Its dimension $\leq d$ is the dimension of its affine linear span.

- point: dimension 0
- segment: dimension 1
- disc or square: dimension 2
- cube or spherical ball: dimension 3

Unfortunately, the spectrahedron is not necessarily full dimensional ($=$ is flat).
It is full dimensional iff it has nonempty interior.
In many practical cases, it has empty interior.

Geometrical degeneracy

For problems with an empty interior, numerical solving fails to produce a checkable solution.

For the most reliable methods,, converge to an approximate solution F: a few very small negative eigenvalues (-10^{-7} or so on small examples).

Articles on exact sdp solving and sums-of-squares generally assume "strict feasibility"!

The problem is easy if strictly feasible

Compute $\tilde{\lambda}$ such that $-F_0 + \sum_i \tilde{\lambda}_i F_i \succeq 0$ according to interior point numerical solving. Interior point solving tends to "push away" from the boundaries and give large eigenvalues. $\Rightarrow$

There is a ball around $\tilde{\lambda}$ where all matrices are sdp.

Method: round $\tilde{F} = -F_0 + \sum_i \tilde{\lambda}_i F_i$ to a nearby rational matrix $F = -F_0 + \sum_i \lambda_i F_i$ (round $\tilde{\lambda}$ to λ is a simple way), check that $F \succeq 0$ in exact arithmetic (Gaussian reduction).

This is basically Parrilo & Weyl's method.

Pipedream

We have a problem $-F_0 + \sum_{i=1}^{d} \lambda_i F_i \succeq 0$ with a degenerate spectrahedron.

If we knew the linear affine span of the spectrahedron, we could reparametrize and obtain a non-degenerate problem $-F_0 + \sum_{i=1}^{d'} \lambda_i F'_i \succeq 0$ in a lower dimension $d' < d$.

How can we know it?

A simple lemma

The nullspace of any matrix in the relative interior of the solution set determines the affine span.

Chicken and egg: get a solution, compute the nullspace, compute the affine span, reparametrize. . . to get a solution

Method: get an approximate solution, compute a reasonable approximate nullspace, etc.

Computing the nullspace

Suppose we have an approximate numerical solution $\tilde{F} = -F_0 + \sum_i \tilde{\lambda}_i F_i$ "almost" $\succeq 0$ and "close" to an exact rational solution F.

Then for all vector v in $\ker F$, $\tilde{F}.v$ is very small.

Bold assumption: $\ker F$ has a basis of small integer vectors.
Then: look for "small" v such that $\tilde{F}.v$ is very small.

Do it by LLL (Lenstra - Lenstra - Lovasz) lattice reduction.

Algorithm

Repeat until success or failure:

- Solve $\tilde{F} = -F_0 + \sum_{i=1}^{d} \tilde{\lambda}_i F_i \succeq 0$ numerically.
- If failed, answer "failure".
- Round $\tilde{\lambda}_i$ to $\lambda_i \in \mathbb{Q}$.
- If $-F_0 + \sum_i \lambda_i F_i \succeq 0$ exactly, answer "success" and print out the λ_i.
- Otherwise, compute some short vectors v such that $\tilde{F}.v$ is small.
- Add the constraint that the solution matrix F should satisfy $F.v = 0$ to the system, obtain a lower dimension problem.

On Positivstellensatz proofs

$$\begin{cases} P_1 = x^3 + xy + 3y^2 + z + 1 \geq 0 \\ P_2 = 5z^3 - 2y^2 + x + 2 \geq 0 \\ P_3 = x^2 + y - z \geq 0 \\ P_4 = -5x^2z^3 - 50xyz^3 - 125y^2z^3 + 2x^2y^2 + 20xy^3 \\ \quad +50y^4 - 2x^3 - 10x^2y - 25xy^2 - 15z^3 - 4x^2 \\ \quad -21xy - 47y^2 - 3x - y - 8 \geq 0 \end{cases}$$

has no solution, but Mathematica 5 (cylindrical algebraic decomposition) cannot prove it, neither can Redlog and QepCad.
Mathematica 7 can but provides no witness.

Our method quickly finds a witness.

Proving impossibilities

14 problems from John Harrison, e.g.

$$0 \leq x \wedge 0 \leq y \wedge 0 \leq z \wedge (xyz = 1) \implies x + y + z \leq x^2 z + y^2 x + z^2 y \quad (1)$$

John's sdp reduction does not converge due to degeneracy, ours converges.

Proving nonnegativity

Took examples in the literature on nonnegative polynomials

- known not to be sums of squares of polynomials
- known to be nonnegative

Compute exact witnesses that they were quotients of sums of squares it at most 7 minutes.

Only one polynomial resisted — large degree, large coefficients, our implementation is too slow.

Coq proof extract

P nonnegative polynomial because $P = N/D$, N and D sums of squares.

```
Definition num:=eval_SOS num_decomp.
Definition denum:=eval_SOS denum_decomp.

Lemma non_critical : forall p q r:Q, r*q == p → p≥
    0 → q≥0 → ~ q == 0 →r ≥0.

Lemma Ident: poly * denum == num.
unfold poly, denum,denum_decomp, num,num_decomp.
simpl eval_SOS. ring.
Qed.

Lemma T:~denum == 0 → poly ≥0.
intros.
apply non_critical with num denum.
apply Ident. apply pos_SOS. apply pos_SOS.
assumption.
Qed.
```

Open problems

Under which conditions of precision does this method converge? How do I set the "scaling factors" used for LLL?

Delicate question: there exist sdp problems that have rational solutions, but none in the relative interior. Thus the assumption that "there is a nearby rational solution" is false in general.

Is it possible to obtain such bad problems from a $\sum_i Q_i P_i = R$ equation ($Q_i \succeq 0$)?

So far

Naive implementation in Sage (Python-based math program).
Post-processing to Coq proofs.

Scalability issues.

Recent work: generate the nullspace faster (use multiple matrices in LLL).