SOS Certificate Repair via DSOS and Rational LP

University of Colorado Boulder

Rameez Wajid Sriram Sankaranarayanan

University of Colorado Boulder

CUPLV Programming Languages and Verification University of Colorado Boulder

Barrier setting: what is the certificate about?

Positivstellensatz certificate template (CBF-style)

Let $K = \{x \mid g_i(x) \geq 0,\ i = 1, \dots, N\}$ be a basic semialgebraic set. For a polynomial barrier $B(x)$, we certify (schematically):

$$\text{(Interior)}\quad B(x) \equiv \sum_i \alpha_i(x)\, p_i(x) + \alpha_0(x)$$

$$\text{(Boundary)}\quad -B(x) \equiv \sum_i \beta_i(x)\, q_i(x) + \beta_0(x)$$

$$\text{(Invariance)}\quad -\nabla B(x)\cdot f(x) + \lambda B(x) \equiv \sum_i \sigma_i(x)\, r_i(x) + \sigma_0(x)$$

where $\alpha_0, \beta_0, \sigma_0$ are SOS and p_i, q_i, r_i are products of the g_i's.

$$\alpha_i, \beta_i, \sigma_i, \dots \implies m(x)^\top Q_i m(x)$$

Q_i should be positive semi-definite

Failure modes (what breaks in practice)

Four common ways a "successful" SOS solution fails to become a proof:

1. **PSD non-rationalizability:** Q is PSD but irrational; rounding breaks identity.
2. **Slight indefiniteness:** $\lambda_{\min}(Q) \approx 0$ (or slightly negative); formal proof needs strict PSD.
3. **Identity drift after rounding:** most common; residue $(x) \neq 0$ over $[x]$.
4. **Tight margins:** near-active constraints produce near-singular Grams; rounding amplifies drift.

SOS Encoding

We introduce **repair slacks** Σ_j constrained to DSOS/SDSOS:

$$\mathrm{LHS}_j(\widehat{B}) - \mathrm{RHS}_j(\widehat{R}_j) - \Sigma_j(x) \equiv 0, \quad j \in \{1, 2, 3\}.$$

Concretely:

$$B \equiv \sum \alpha_i p_i + \alpha_0 + \alpha_{\mathrm{DSOS/SDSOS}},$$
$$-B \equiv \sum \beta_i q_i + \beta_0 + \beta_{\mathrm{DSOS/SDSOS}},$$
$$-\nabla B \cdot f + \lambda B \equiv \sum \sigma_i r_i + \sigma_0 + \sigma_{\mathrm{DSOS/SDSOS}}.$$

$$\alpha_i, \beta_i, \sigma_i, \dots \implies m(x)^\top Q_i m(x)$$

Q_i should be positive semi-definite

Proof-producing SOS barrier pipeline

SOS solve → Naive rounding → Residue → DSOS repair → Exact LP over $\mathbb{Q}$ → Final certificate

SOS solve. Solve an SDP in double precision to obtain Q_{fp}:

$$p(x) = z(x)^\top Q_{\mathrm{fp}} z(x), \qquad Q_{\mathrm{fp}} \succeq 0$$

(numerically, up to solver tolerances).

Naive rationalization. Entrywise rounding $Q_{\mathrm{rat}} = \mathrm{round}(Q_{\mathrm{fp}})$ gives rationals, but small perturbations can make

$$\lambda_{\min}(Q_{\mathrm{rat}}) < 0,$$

so Q_{rat} is no longer a valid SOS Gram matrix.

Residue as data. Track the error polynomial

$$r(x) = p(x) - z(x)^\top Q_{\mathrm{rat}} z(x).$$

This small but non-zero residue becomes the target for a repair step.

DSOS repair. Linear program in Gram entries:

$$z(x)^\top Q_{\mathrm{corr}} z(x) = p(x)$$

with diagonal dominance constraints, so $Q_{\mathrm{corr}} \succeq 0$. Solved with an LP solver (e.g. GLPK, Gurobi).

Exact LP over $\mathbb{Q}$. Re-solve the DSOS LP using rational arithmetic (CDDLib in Julia) to obtain

$$Q_{\mathbb{Q}} \in \mathbb{Q}^{n\times n}, \quad z(x)^\top Q_{\mathbb{Q}} z(x) = p(x).$$

This removes all floating-point error from the certificate.

Final certificate. Factor $Q_{\mathbb{Q}} = L^\top L$ (LDL$^\top$/Cholesky) and export

$$p(x) = \sum_k \ell_k(x)^2$$

as a machine-checkable artifact (Lean, Coq).

Takeaway: SOS is powerful but numerical; DSOS repair + exact LP turns it into an auditable proof.

" From floating-point witnesses to exact, checkable proofs. "

rameez.wajid@colorado.edu