Interpolant Synthesis for Quadratic Polynomial Inequalities and Combination with *EUF*

Deepak Kapur

Department of Computer Science, University of New Mexico

Joint work with Ting Gan, Liyun Dai, Bican Xia, Naijun Zhan, and Mingshuai Chen

Dagstuhl, September 2015

Outline

1. Key ideas

Outline

Outline

1. Key ideas
   - Generalization of Motzkin's transposition theorem
   - Concave quadratic polynomials

Outline

1. Key ideas
   - Generalization of Motzkin's transposition theorem
   - Concave quadratic polynomials
   - Positive constant replaced by sum of squares

Outline

1. Key ideas
   - Generalization of Motzkin's transposition theorem
   - Concave quadratic polynomials
   - Positive constant replaced by sum of squares
2. Generating interpolants for Concave Quadratic Polynomial inequalities

Outline

1. Key ideas
   - Generalization of Motzkin's transposition theorem
   - Concave quadratic polynomials
   - Positive constant replaced by sum of squares
2. Generating interpolants for Concave Quadratic Polynomial inequalities
   - NSOSC condition : generalized Motzkin's theorem applies

Outline

1. Key ideas
   - Generalization of Motzkin's transposition theorem
   - Concave quadratic polynomials
   - Positive constant replaced by sum of squares
2. Generating interpolants for Concave Quadratic Polynomial inequalities
   - $\mathbf{NSOSC}$ condition : generalized Motzkin's theorem applies
   - $\mathbf{SOS}$ ($\mathbf{NSOSC}$ not satisifed) : equalities from expressions in a sum of squares being equal to 0.

Outline

1. Key ideas
   - Generalization of Motzkin's transposition theorem
   - Concave quadratic polynomials
   - Positive constant replaced by sum of squares
2. Generating interpolants for Concave Quadratic Polynomial inequalities
   - $\mathbf{NSOSC}$ condition : generalized Motzkin's theorem applies
   - $\mathbf{SOS}$ ($\mathbf{NSOSC}$ not satisifed) : equalities from expressions in a sum of squares being equal to 0.
3. Combination with uninterpreted function symbols (*EUF*)

Outline

1. Key ideas
   - Generalization of Motzkin's transposition theorem
   - Concave quadratic polynomials
   - Positive constant replaced by sum of squares
2. Generating interpolants for Concave Quadratic Polynomial inequalities
   - $\mathbf{NSOSC}$ condition : generalized Motzkin's theorem applies
   - $\mathbf{SOS}$ ($\mathbf{NSOSC}$ not satisifed) : equalities from expressions in a sum of squares being equal to 0.
3. Combination with uninterpreted function symbols (*EUF*)
   - similar to the linear case

Outline

1. Key ideas
   - Generalization of Motzkin's transposition theorem
   - Concave quadratic polynomials
   - Positive constant replaced by sum of squares
2. Generating interpolants for Concave Quadratic Polynomial inequalities
   - $\mathbf{NSOSC}$ condition : generalized Motzkin's theorem applies
   - $\mathbf{SOS}$ ($\mathbf{NSOSC}$ not satisifed) : equalities from expressions in a sum of squares being equal to 0.
3. Combination with uninterpreted function symbols (*EUF*)
   - similar to the linear case
4. Concluding remarks

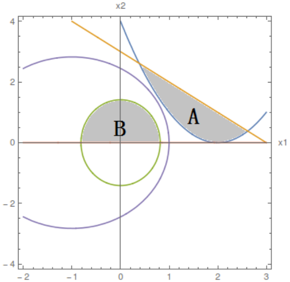
Overview of the idea

Example (running example)

Consider two formulas A and B with $A \wedge B \models \perp$, where

$$A := -{x_1}^2 + 4x_1 + x_2 - 4 \geq 0 \wedge -x_1 - x_2 + 3 - y^2 > 0,$$
$$B := -3{x_1}^2 - {x_2}^2 + 1 \geq 0 \wedge x_2 - z^2 \geq 0$$

We aim to generate an interpolant I for A and B, on the common variables (x_1 and x_2), such that $A \models I$ and $I \wedge B \models \perp$.



An intuitive description of a candidate interpolant is as the purple curve in the above right figure, which separates A and B in the panel of x_1 and x_2.

Overview of the idea

- A polynomial time algorithm for generating interpolants from mutually contradictory conjunctions of concave quadratic polynomial inequalities over the reals :

Overview of the idea

- A polynomial time algorithm for generating interpolants from mutually contradictory conjunctions of concave quadratic polynomial inequalities over the reals :
  - If no nonpositive constant combination of nonstrict inequalities is a sum of squares polynomial, an interpolant a la McMillan can be generated essentially using the linearization of quadratic polynomials.

Overview of the idea

- A polynomial time algorithm for generating interpolants from mutually contradictory conjunctions of concave quadratic polynomial inequalities over the reals :
  - If no nonpositive constant combination of nonstrict inequalities is a sum of squares polynomial, an interpolant a la McMillan can be generated essentially using the linearization of quadratic polynomials.
  - Otherwise, linear equalities relating variables are deduced, resulting to interpolation subproblems with fewer variables on which the algorithm is recursively applied.

Overview of the idea

- A polynomial time algorithm for generating interpolants from mutually contradictory conjunctions of concave quadratic polynomial inequalities over the reals :
  - If no nonpositive constant combination of nonstrict inequalities is a sum of squares polynomial, an interpolant a la McMillan can be generated essentially using the linearization of quadratic polynomials.
  - Otherwise, linear equalities relating variables are deduced, resulting to interpolation subproblems with fewer variables on which the algorithm is recursively applied.
- An algorithm for generating interpolants for the combination of quantifier-free theory of concave quadratic polynomial inequalities and equality theory over uninterpreted function symbols (*EUF*).

Preliminaries

Theorem (Motzkin's transposition theorem)

Let A and B be matrices and let $\vec{\alpha}$ and $\vec{\beta}$ be column vectors. Then there exists a vector $\mathbf{x}$ with $A\mathbf{x} - \vec{\alpha} \geq 0$ and $B\mathbf{x} - \vec{\beta} > 0$, iff for all row vectors $\mathbf{y}, \mathbf{z} \geq 0$:

$$(i) \text{ if } \mathbf{y}A + \mathbf{z}B = 0 \text{ then } \mathbf{y}\vec{\alpha} + \mathbf{z}\vec{\beta} \leq 0;$$
$$(ii) \text{ if } \mathbf{y}A + \mathbf{z}B = 0 \text{ and } \mathbf{z} \neq 0 \text{ then } \mathbf{y}\vec{\alpha} + \mathbf{z}\vec{\beta} < 0.$$

Preliminaries

Theorem (Motzkin's transposition theorem)

Let A and B be matrices and let $\vec{\alpha}$ and $\vec{\beta}$ be column vectors. Then there exists a vector $\mathbf{x}$ with $A\mathbf{x} - \vec{\alpha} \geq 0$ and $B\mathbf{x} - \vec{\beta} > 0$, iff for all row vectors $\mathbf{y}, \mathbf{z} \geq 0$:

$$
\begin{aligned}
(i)\ &\text{if } \mathbf{y}A + \mathbf{z}B = 0 \text{ then } \mathbf{y}\vec{\alpha} + \mathbf{z}\vec{\beta} \leq 0; \\
(ii)\ &\text{if } \mathbf{y}A + \mathbf{z}B = 0 \text{ and } \mathbf{z} \neq 0 \text{ then } \mathbf{y}\vec{\alpha} + \mathbf{z}\vec{\beta} < 0.
\end{aligned}
$$

Corollary

Let $A \in \mathbb{R}^{r \times n}$ and $B \in \mathbb{R}^{s \times n}$ be matrices and $\vec{\alpha} \in \mathbb{R}^r$ and $\vec{\beta} \in \mathbb{R}^s$ be column vectors, where $A_i, i = 1, \ldots, r$ is the ith row of A and $B_j, j = 1, \ldots, s$ is the jth row of B. There does not exist a vector $\mathbf{x}$ with $A\mathbf{x} - \vec{\alpha} \geq 0$ and $B\mathbf{x} - \vec{\beta} > 0$, iff there exist real numbers $\lambda_1, \ldots, \lambda_r \geq 0$ and $\eta_0, \eta_1, \ldots, \eta_s \geq 0$ such that

$$
\sum_{i=1}^{r} \lambda_i (A_i \mathbf{x} - \alpha_i) + \sum_{j=1}^{s} \eta_j (B_j \mathbf{x} - \beta_j) + \eta_0 \equiv 0 \;\; \textit{with} \sum_{j=0}^{s} \eta_j = 1. \tag{1}
$$

Concave quadratic polynomials

Definition (Concave Quadratic)

A polynomial $f \in \mathbb{R}[\mathbf{x}]$ is called *concave quadratic (CQ)* if the following two conditions hold :

- f has total degree at most 2, i.e., it has the form $f = \mathbf{x}^T A \mathbf{x} + 2\vec{\alpha}^T \mathbf{x} + a$, where A is a real symmetric matrix, $\vec{\alpha}$ is a column vector and $a \in \mathbb{R}$;
- the matrix A is negative semi-definite, written as $A \preceq 0$.

Example

Take $f = -3{x_1}^2 - {x_2}^2 + 1$ in the running example, which is from the ellipsoid domain and can be expressed as

$$f = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^T \begin{pmatrix} -3 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + 1.$$

The corresponding $A = \begin{pmatrix} -3 & 0 \\ 0 & -1 \end{pmatrix} \preceq 0$. Thus, f is CQ.

Concave quadratic polynomials

- If $f \in \mathbb{R}[\mathbf{x}]$ is linear, then f is CQ because its total degree is 1 and the corresponding A is 0 which is of course negative semi-definite.

Concave quadratic polynomials

- If $f \in \mathbb{R}[\mathbf{x}]$ is linear, then f is CQ because its total degree is 1 and the corresponding A is 0 which is of course negative semi-definite.
- A quadratic polynomial $f(\mathbf{x}) = \mathbf{x}^T A\mathbf{x} + 2\vec{\alpha}^T\mathbf{x} + a$ can also be represented as an inner product of matrices, i.e., $\left\langle P, \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{x}\mathbf{x}^T \end{pmatrix} \right\rangle$, where $P = \begin{pmatrix} a & \alpha^T \\ \alpha & A \end{pmatrix}$.

Linearization of CQ polynomials

Definition (Linearization)

Given a quadratic polynomial $f(\mathbf{x}) = \left\langle P, \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{x}\mathbf{x}^T \end{pmatrix} \right\rangle$, its *linearization* is defined as $f(\mathbf{x}) = \left\langle P, \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \vec{X} \end{pmatrix} \right\rangle$, where $\begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \vec{X} \end{pmatrix} \succeq 0$.

Linearization of CQ polynomials

Definition (Linearization)

Given a quadratic polynomial $f(\mathbf{x}) = \left\langle P, \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{x}\mathbf{x}^T \end{pmatrix} \right\rangle$, its *linearization* is defined as $f(\mathbf{x}) = \left\langle P, \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \vec{X} \end{pmatrix} \right\rangle$, where $\begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \vec{X} \end{pmatrix} \succeq 0$.

let

$$K \quad \hat{=} \quad \{\mathbf{x} \in \mathbb{R}^n \mid f_1(\mathbf{x}) \geq 0, \ldots, f_r(\mathbf{x}) \geq 0, g_1(\mathbf{x}) > 0, \ldots, g_s(\mathbf{x}) > 0\}, \tag{2}$$

$$K_1 \quad \hat{=} \quad \{\mathbf{x} \mid \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \vec{X} \end{pmatrix} \succeq 0, \ \wedge_{i=1}^{r} \left\langle P_i, \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \vec{X} \end{pmatrix} \right\rangle \geq 0,$$

$$\wedge_{j=1}^{s} \left\langle Q_j, \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \vec{X} \end{pmatrix} \right\rangle > 0, \text{ for some } \vec{X}\}, \tag{3}$$

Linearization of CQ polynomials

Theorem

Let $f_1, \ldots, f_r$ and $g_1, \ldots, g_s$ be CQ polynomials, K and K_1 as above, then $K = K_1$.

Therefore, when f_is and g_js are CQ, the CQ polynomial inequalities can be transformed equivalently to a set of linear inequality constraints and a positive semi-definite constraint.

Problem formulation

Problem 1

Given two formulas ϕ and ψ on n variables with $\phi \wedge \psi \models \bot$, where

$$
\begin{aligned}
\phi &= f_1 \geq 0 \wedge \ldots \wedge f_{r_1} \geq 0 \wedge g_1 > 0 \wedge \ldots \wedge g_{s_1} > 0, \\
\psi &= f_{r_1+1} \geq 0 \wedge \ldots \wedge f_r \geq 0 \wedge g_{s_1+1} > 0 \wedge \ldots \wedge g_s > 0,
\end{aligned}
$$

in which $f_1, \ldots, f_r, g_1, \ldots, g_s$ are all CQ, develop an algorithm to generate a (reverse) Craig interpolant I for ϕ and ψ, on the common variables of ϕ and ψ, such that $\phi \models I$ and $I \wedge \psi \models \bot$.

$\mathbf{x} = (x_1, \ldots, x_d)$, $\mathbf{y} = (y_1, \ldots, y_u)$ and $\mathbf{z} = (z_1, \ldots, z_v)$, where $d + u + v = n$.

NSOSC Condition

Definition (NSOSC)

Formulas ϕ and ψ in Problem 1, satisfy the non-existence of an SOS polynomial condition (NSOSC) iff there do not exist $\delta_1 \geq 0, \ldots, \delta_r \geq 0$, s.t. $-(\delta_1 f_1 + \ldots + \delta_r f_r)$ is a non-zero SOS.

Example

Formulas A and B in the running example do not satisfy NSOSC, since there exist $\delta_1 = 1, \delta_2 = 1, \delta_3 = 1$, s.t.

$$
\begin{aligned}
& -(\delta_1(-{x_1}^2 + 4x_1 + x_2 - 4) + \delta_2(-3{x_1}^2 - {x_2}^2 + 1) + \delta_3(x_2 - z^2)) \\
& = (2x_1 - 1)^2 + (x_2 - 1)^2 + z^2
\end{aligned}
$$

is a non-zero SOS.

Generalization of Motzkin's theorem

Theorem (Generalization of Motzkin's theorem)

Let $f_1, \ldots, f_r, g_1, \ldots, g_s$ be CQ polynomials whose conjunction is unsatisfiable. If the condition $\mathbf{NSOSC}$ holds, then there exist $\lambda_i \geq 0$ ($i = 1, \cdots, r$), $\eta_j \geq 0$ ($j = 0, 1, \cdots, s$) and a quadratic SOS polynomial h of the form $(l_1)^2 + \ldots + (l_k)^2$ where l_i are linear expressions in x, y, z. , s.t.

$$\sum_{i=1}^{r} \lambda_i f_i + \sum_{j=1}^{s} \eta_j g_j + \eta_0 + h \equiv 0, \tag{4}$$

$$\eta_0 + \eta_1 + \ldots + \eta_s = 1. \tag{5}$$

Using this generalization, an interpolant for ϕ and ψ is generated from the SOS polynomial h by splitting it into two SOS polynomials.

When NSOSC is satisfied

Theorem

Let ϕ and ψ as defined in Problem 1 with $\phi \wedge \varphi \models \perp$, which satisfy NSOSC. Then there exist $\lambda_i \geq 0$ ($i = 1, \cdots, r$), $\eta_j \geq 0$ ($j = 0, 1, \cdots, s$) and two quadratic SOS polynomial $h_1 \in \mathbb{R}[\mathbf{x}, \mathbf{y}]$ and $h_2 \in \mathbb{R}[\mathbf{x}, \mathbf{z}]$ s.t.

$$\sum_{i=1}^{r} \lambda_i f_i + \sum_{j=1}^{s} \eta_j g_j + \eta_0 + h_1 + h_2 \equiv 0, \tag{6}$$

$$\eta_0 + \eta_1 + \ldots + \eta_s = 1. \tag{7}$$

Let $I = \sum_{i=1}^{r_1} \lambda_i f_i + \sum_{j=1}^{s_1} \eta_j g_j + \eta_0 + h_1 \in \mathbb{R}[\mathbf{x}]$. Then, if $\sum_{j=0}^{s_1} \eta_j > 0$, then $I > 0$ is an interpolant ; otherwise $I \geq 0$ is an interpolant.

Computing interpolant using semi-definite programming

Let $W = \begin{pmatrix} 1 & \mathbf{x}^T & \mathbf{y}^T & \mathbf{z}^T \\ \mathbf{x} & \mathbf{x}\mathbf{x}^T & \mathbf{x}\mathbf{y}^T & \mathbf{x}\mathbf{z}^T \\ \mathbf{y} & \mathbf{y}\mathbf{x}^T & \mathbf{y}\mathbf{y}^T & \mathbf{y}\mathbf{z}^T \\ \mathbf{z} & \mathbf{z}\mathbf{x}^T & \mathbf{z}\mathbf{y}^T & \mathbf{z}\mathbf{z}^T \end{pmatrix}$, $f_i = \langle P_i, W \rangle, \;\; g_j = \langle Q_j, W \rangle$, where P_i and Q_j are $(n+1) \times (n+1)$ matrices, $h_1 = \langle M, W \rangle, \;\; h_2 = \langle \hat{M}, W \rangle$, and $M = (M_{ij})_{4\times 4}$, $\hat{M} = (\hat{M}_{ij})_{4\times 4}$ with appropriate dimensions, e.g., $M_{12} \in \mathbb{R}^{1\times d}$ and $\hat{M}_{34} \in \mathbb{R}^{u\times v}$.

Then, with $\mathbf{NSOSC}$, computing the interpolant is reduced to the following $\mathbf{SDP}$ feasibility problem :

Computing interpolant using semi-definite programming

Find : $\lambda_1, \ldots, \lambda_r, \eta_1, \ldots, \eta_s \in \mathbb{R}$, $M, \hat{M} \in \mathbb{R}^{(n+1)\times(n+1)}$ subject to

$$\begin{cases} \sum_{i=1}^{r} \lambda_i P_i + \sum_{j=1}^{s} \eta_j Q_j + \eta_0 E_{1,1} + M + \hat{M} = 0, \sum_{j=1}^{s} \eta_j = 1, \\ M_{41} = (M_{14})^T = 0, M_{42} = (M_{24})^T = 0, M_{43} = (M_{34})^T = 0, M_{44} = 0, \\ \hat{M}_{31} = (\hat{M}_{13})^T = 0, \hat{M}_{32} = (\hat{M}_{23})^T = 0, \hat{M}_{33} = 0, \hat{M}_{34} = (\hat{M}_{43})^T = 0, \\ M \succeq 0, \hat{M} \succeq 0, \lambda_i \geq 0, \eta_j \geq 0, \text{ for } i = 1, \ldots, r, j = 1, \ldots, s, \end{cases}$$

where $E_{(1,1)}$ is a $(n+1) \times (n+1)$ matrix, whose all other entries are 0 except for $(1,1)$ entry being 1.

Generating interpolants when NSOSC holds

Algorithm 1: IGFCH

input : Two formulas ϕ, ψ with **NSOSC** and $\phi \wedge \psi \models \perp$, where
$\phi = f_1 \geq 0 \wedge \ldots \wedge f_{r_1} \geq 0 \wedge g_1 > 0 \wedge \ldots \wedge g_{s_1} > 0$,
$\psi = f_{r_1+1} \geq 0 \wedge \ldots \wedge f_r \geq 0 \wedge g_{s_1+1} > 0 \wedge \ldots \wedge g_s > 0$,
$f_1, \ldots, f_r, g_1, \ldots, g_s$ are all concave quadratic polynomials,
$f_1, \ldots, f_{r_1}, g_1, \ldots, g_{s_1} \in \mathbb{R}[\mathbf{x}, \mathbf{y}]$, $f_{r_1+1}, \ldots, f_r, g_{s_1+1}, \ldots, g_s \in \mathbb{R}[\mathbf{x}, \mathbf{z}]$

output: A formula I to be an interpolant for ϕ and ψ

Find $\lambda_1, \ldots, \lambda_r \geq 0, , \eta_0, \eta_1, \ldots, \eta_s \geq 0, h_1 \in \mathbb{R}[\mathbf{x}, \mathbf{y}], h_2 \in \mathbb{R}[\mathbf{x}, \mathbf{z}]$ by SDP s.t.

$$\sum_{i=1}^{r} \lambda_i f_i + \sum_{j=1}^{s} \eta_j g_j + \eta_0 + h_1 + h_2 \equiv 0,$$

$$\eta_0 + \eta_1 + \ldots + \eta_s = 1,$$

h_1, h_2 are SOS polynomials;

/* This is essentially a **SDP** problem, see Section 4.2 */

$f := \sum_{i=1}^{r_1} \lambda_i f_i + \sum_{j=1}^{s_1} \eta_j g_j + \eta_0 + h_1$;

if $\sum_{j=0}^{s_1} \eta_j > 0$ **then** $I := (f > 0)$; **else** $I := (f \geq 0)$;

return I

When NSOSC is not satisfied

If ϕ and ψ do not satisfy $\mathbf{NSOSC}$, i.e., an SOS polynomial $h(\mathbf{x}, \mathbf{y}, \mathbf{z}) = -(\sum_{i=1}^{r} \lambda_i f_i)$ can be computed which can be split into two SOS polynomials $h_1(\mathbf{x}, \mathbf{y})$ and $h_2(\mathbf{x}, \mathbf{z})$ as discussed previously. Then an SOS polynomial $f(\mathbf{x})$ such that $\phi \models f(\mathbf{x}) \geq 0$ and $\psi \models -f(\mathbf{x}) \geq 0$ can be constructed as

$$f(\mathbf{x}) = (\sum_{i=1}^{r_1} \delta_i f_i) + h_1 = -(\sum_{i=r_1+1}^{r} \delta_i f_i) - h_2, \delta_i \geq 0.$$

Lemma

If Problem 1 does not satisfy $\mathbf{NSOSC}$*, there exists* $f \in \mathbb{R}[\mathbf{x}]$*, s.t.* $\phi \Leftrightarrow \phi_1 \vee \phi_2$ *and* $\psi \Leftrightarrow \psi_1 \vee \psi_2$*, where,*

$$\phi_1 = (f > 0 \wedge \phi), \phi_2 = (f = 0 \wedge \phi), \psi_1 = (-f > 0 \wedge \psi), \psi_2 = (f = 0 \wedge \psi). \quad (8)$$

When NSOSC is not satisfied

Using the previous lemma, an interpolant I for ϕ and ψ can be constructed from an interpolant $I_{2,2}$ for ϕ_2 and ψ_2.

Theorem

With $\phi, \psi, \phi_1, \phi_2, \psi_1, \psi_2$ as in previous Lemma, from an interpolant $I_{2,2}$ for ϕ_2 and ψ_2, $I := (f > 0) \vee (f \geq 0 \wedge I_{2,2})$ is an interpolant for ϕ and ψ.

When NSOSC is not satisfied

Using the previous lemma, an interpolant I for ϕ and ψ can be constructed from an interpolant $I_{2,2}$ for ϕ_2 and ψ_2.

Theorem

With $\phi, \psi, \phi_1, \phi_2, \psi_1, \psi_2$ as in previous Lemma, from an interpolant $I_{2,2}$ for ϕ_2 and ψ_2, $I := (f > 0) \vee (f \geq 0 \wedge I_{2,2})$ is an interpolant for ϕ and ψ.

If h and hence h_1, h_2 have a positive constant $a_{n+1} > 0$, then f cannot be 0, implying that ϕ_2, ψ_2 are $\perp$. We thus have :

Theorem

With $\phi, \psi, \phi_1, \phi_2, \psi_1, \psi_2$ as in previous Lemma and h has $a_{n+1} > 0$, $f > 0$ is an interpolant for ϕ and ψ.

In case h does not have a constant, i.e., $a_{n+1} = 0$, elimination of variables can be recursively performed to terminate the algorithm.

Generating interpolants for CQI

Algorithm 2: IGFQC

input : Two formulas ϕ, ψ with $\phi \wedge \psi \models \bot$, where
$\phi = f_1 \geq 0 \wedge \ldots \wedge f_{r_1} \geq 0 \wedge g_1 > 0 \wedge \ldots \wedge g_{s_1} > 0$,
$\psi = f_{r_1+1} \geq 0 \wedge \ldots \wedge f_r \geq 0 \wedge g_{s_1+1} > 0 \wedge \ldots \wedge g_s > 0$,
$f_1, \ldots, f_r, g_1, \ldots, g_s$ are all CQ polynomials,
$f_1, \ldots, f_{r_1}, g_1, \ldots, g_{s_1} \in \mathbb{R}[\mathbf{x}, \mathbf{y}]$, and $f_{r_1+1}, \ldots, f_r, g_{s_1+1}, \ldots, g_s \in \mathbb{R}[\mathbf{x}, \mathbf{z}]$

output: A formula I to be an interpolant for ϕ and ψ

if *Var*$(\phi) \subseteq$ *Var*(ψ) **then** $I := \phi$; **return** I;
Find $\delta_1, \ldots, \delta_r \geq 0, h \in \mathbb{R}[\mathbf{x}, \mathbf{y}, \mathbf{z}]$ by SDP s.t. $\sum_{i=1}^{r} \delta_i f_i + h \equiv 0$ and h is SOS;
/* Check the condition **NSOSC** */
if *no solution* **then** $I :=$ **IGFCH**(ϕ, ψ); **return** I;
/* **NSOSC** holds */
Construct $h_1 \in \mathbb{R}[\mathbf{x}, \mathbf{y}]$ and $h_2 \in \mathbb{R}[\mathbf{x}, \mathbf{z}]$ with the forms (H1) and (H2);
$f := \sum_{i=1}^{r_1} \delta_i f_i + h_1 = -\sum_{i=r_1+1}^{r} \delta_i f_i - h_2$;
Construct ϕ' and ψ' using Theorem 6 and Theorem 7 by eliminating variables due to $h_1 = h_2 = 0$;
$I' =$ **IGFQC**(ϕ', ψ');
$I := (f > 0) \vee (f \geq 0 \wedge I')$;
return I

Generating interpolants for CQI

Example

Recall the running example where

$$\begin{aligned} h &= (2x_1 - 1)^2 + (x_2 - 1)^2 + z^2 \\ &= \underbrace{\frac{1}{2}((2x_1 - 1)^2 + (x_2 - 1)^2)}_{h_1} + \underbrace{\frac{1}{2}((2x_1 - 1)^2 + (x_2 - 1)^2) + z^2}_{h_2} \end{aligned}$$

$$\begin{aligned} f &= \delta_1(-{x_1}^2 + 4x_1 + x_2 - 4) + h_1 \\ &= -3 + 2x_1 + {x_1}^2 + \frac{1}{2}{x_2}^2 \end{aligned}$$

We construct A' from A by setting $x_1 = \frac{1}{2}, x_2 = 1$ derived from $h_1 = 0$; similarly B' is constructed by setting $x_1 = \frac{1}{2}, x_2 = 1, z = 0$ in B as derived from $h_2 = 0$. It follows that, $A' := B' := \perp$ Thus, $I(A', B') := (0 > 0)$ is an interpolant for (A', B').

An interpolant for A and B is thus $(f(x) > 0) \vee (f(x) = 0 \wedge I(A', B'))$, i.e.

$$-3 + 2x_1 + {x_1}^2 + \frac{1}{2}{x_2}^2 > 0.$$

which corresponds to the purple curve mentioned previously.

Combination with EUF

- $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$: a finite set of uninterpreted function symbols in *EUF*;

Combination with EUF

- $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$: a finite set of uninterpreted function symbols in *EUF*;
- $\Omega_{12} = \Omega_1 \cup \Omega_2, \Omega_{13} = \Omega_1 \cup \Omega_3$;

Combination with EUF

- $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$: a finite set of uninterpreted function symbols in *EUF*;
- $\Omega_{12} = \Omega_1 \cup \Omega_2$, $\Omega_{13} = \Omega_1 \cup \Omega_3$;
- $\mathbb{R}[\mathbf{x}, \mathbf{y}, \mathbf{z}]^{\Omega}$: the extension of $\mathbb{R}[\mathbf{x}, \mathbf{y}, \mathbf{z}]$ in which polynomials can have terms built using function symbols in Ω and variables in $\mathbf{x}, \mathbf{y}, \mathbf{z}$.

Combination with EUF

- $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$: a finite set of uninterpreted function symbols in *EUF*;
- $\Omega_{12} = \Omega_1 \cup \Omega_2$, $\Omega_{13} = \Omega_1 \cup \Omega_3$;
- $\mathbb{R}[\mathbf{x}, \mathbf{y}, \mathbf{z}]^{\Omega}$: the extension of $\mathbb{R}[\mathbf{x}, \mathbf{y}, \mathbf{z}]$ in which polynomials can have terms built using function symbols in Ω and variables in $\mathbf{x}, \mathbf{y}, \mathbf{z}$.

Combination with EUF

- $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$: a finite set of uninterpreted function symbols in *EUF*;
- $\Omega_{12} = \Omega_1 \cup \Omega_2$, $\Omega_{13} = \Omega_1 \cup \Omega_3$;
- $\mathbb{R}[\mathbf{x}, \mathbf{y}, \mathbf{z}]^{\Omega}$: the extension of $\mathbb{R}[\mathbf{x}, \mathbf{y}, \mathbf{z}]$ in which polynomials can have terms built using function symbols in Ω and variables in $\mathbf{x}, \mathbf{y}, \mathbf{z}$.

Problem 2

Suppose two formulas ϕ and ψ with $\phi \wedge \psi \models \bot$, where

$$\phi = f_1 \geq 0 \wedge \ldots \wedge f_{r_1} \geq 0 \wedge g_1 > 0 \wedge \ldots \wedge g_{s_1} > 0,$$
$$\psi = f_{r_1+1} \geq 0 \wedge \ldots \wedge f_r \geq 0 \wedge g_{s_1+1} > 0 \wedge \ldots \wedge g_s > 0,$$

where $f_1, \ldots, f_r, g_1, \ldots, g_s$ are all CQ polynomials, $f_1, \ldots, f_{r_1}, g_1, \ldots, g_{s_1} \in \mathbb{R}[\mathbf{x}, \mathbf{y}]^{\Omega_{12}}$, $f_{r_1+1}, \ldots, f_r, g_{s_1+1}, \ldots, g_s \in \mathbb{R}[\mathbf{x}, \mathbf{z}]^{\Omega_{13}}$, the goal is to generate an interpolant I for ϕ and ψ, expressed using the common symbols $\mathbf{x}, \Omega_1$, i.e., I includes only polynomials in $\mathbb{R}[\mathbf{x}]^{\Omega_1}$.

Sketch of the idea (Algorithm $\mathbf{IGFQCE}$)

1. Flatten and purify the formulas ϕ and ψ as $\overline{\phi}$ and $\overline{\psi}$ by introducing fresh variables for each term with uninterpreted symbols as well as for the terms with uninterpreted symbols.

Sketch of the idea (Algorithm IGFQCE)

1. Flatten and purify the formulas ϕ and ψ as $\overline{\phi}$ and $\overline{\psi}$ by introducing fresh variables for each term with uninterpreted symbols as well as for the terms with uninterpreted symbols.
2. Generate a set N of Horn clauses as

   $$N = \{\bigwedge_{k=1}^{n} c_k = b_k \rightarrow c = b \mid \omega(c_1, \dots, c_n) = c \in D, \omega(b_1, \dots, b_n) = b \in D\},$$

   where D consists of unit clauses of the form $\omega(c_1, \dots, c_n) = c$, with $c_1, \dots, c_n$ be variables and $\omega \in \Omega$.

Sketch of the idea (Algorithm IGFQCE)

1. Flatten and purify the formulas ϕ and ψ as $\overline{\phi}$ and $\overline{\psi}$ by introducing fresh variables for each term with uninterpreted symbols as well as for the terms with uninterpreted symbols.
2. Generate a set N of Horn clauses as
$$N = \{\bigwedge_{k=1}^{n} c_k = b_k \to c = b \mid \omega(c_1, \ldots, c_n) = c \in D, \omega(b_1, \ldots, b_n) = b \in D\},$$
where D consists of unit clauses of the form $\omega(c_1, \ldots, c_n) = c$, with $c_1, \ldots, c_n$ be variables and $\omega \in \Omega$.
3. Partition N into N_ϕ, N_ψ, and N_{mix} with all symbols in N_ϕ, N_ψ appearing in $\overline{\phi}$, $\overline{\psi}$, respectively, and N_{mix} consisting of symbols from both $\overline{\phi}, \overline{\psi}$.
$$\phi \wedge \psi \models \bot \text{ iff } \overline{\phi} \wedge \overline{\psi} \wedge D \models \bot \text{ iff } (\overline{\phi} \wedge N_\phi) \wedge (\overline{\psi} \wedge N_\psi) \wedge N_{\text{mix}} \models \bot. \quad (9)$$

Sketch of the idea (Algorithm $\mathbf{IGFQCE}$)

1. Flatten and purify the formulas ϕ and ψ as $\overline{\phi}$ and $\overline{\psi}$ by introducing fresh variables for each term with uninterpreted symbols as well as for the terms with uninterpreted symbols.
2. Generate a set N of Horn clauses as

   $$N = \{\bigwedge_{k=1}^{n} c_k = b_k \rightarrow c = b \mid \omega(c_1, \ldots, c_n) = c \in D, \omega(b_1, \ldots, b_n) = b \in D\},$$

   where D consists of unit clauses of the form $\omega(c_1, \ldots, c_n) = c$, with $c_1, \ldots, c_n$ be variables and $\omega \in \Omega$.
3. Partition N into N_ϕ, N_ψ, and N_{mix} with all symbols in N_ϕ, N_ψ appearing in $\overline{\phi}$, $\overline{\psi}$, respectively, and N_{mix} consisting of symbols from both $\overline{\phi}, \overline{\psi}$.

   $$\phi \wedge \psi \models \bot \text{ iff } \overline{\phi} \wedge \overline{\psi} \wedge D \models \bot \text{ iff } (\overline{\phi} \wedge N_\phi) \wedge (\overline{\psi} \wedge N_\psi) \wedge N_{\text{mix}} \models \bot. \quad (9)$$

4. Generate interpolant : Notice that $(\overline{\phi} \wedge N_\phi) \wedge (\overline{\psi} \wedge N_\psi) \wedge N_{\text{mix}} \models \bot$ has no uninterpreted function symbols. If N_{mix} can be replaced by N_{sep}^{ϕ} and N_{sep}^{ψ} as in [Rybalchenko & Sofronie-Stokkermans 10] using separating terms, then $\mathbf{IGFQC}$ can be applied. An interpolant generated for this problem can be used to generate an interpolant for ϕ, ψ after uniformly replacing all new symbols by their corresponding expressions from D.

An illustrating example

Example

$$\begin{aligned}\phi :=& (f_1 = -(y_1 - x_1 + 1)^2 - x_1 + x_2 \geq 0) \wedge (y_2 = \alpha(y_1) + 1) \\ & \wedge (g_1 = -x_1^2 - x_2^2 - y_2^2 + 1 > 0), \\ \psi :=& (f_2 = -(z_1 - x_2 + 1)^2 + x_1 - x_2 \geq 0) \wedge (z_2 = \alpha(z_1) - 1) \\ & \wedge (g_2 = -x_1^2 - x_2^2 - z_2^2 + 1 > 0).\end{aligned}$$

An illustrating example

Example

$$\phi := (f_1 = -(y_1 - x_1 + 1)^2 - x_1 + x_2 \geq 0) \land (y_2 = \alpha(y_1) + 1)$$
$$\land\, (g_1 = -x_1^2 - x_2^2 - y_2^2 + 1 > 0),$$
$$\psi := (f_2 = -(z_1 - x_2 + 1)^2 + x_1 - x_2 \geq 0) \land (z_2 = \alpha(z_1) - 1)$$
$$\land\, (g_2 = -x_1^2 - x_2^2 - z_2^2 + 1 > 0).$$

1. Flattening and purification gives

$$\overline{\phi} := (f_1 \geq 0 \land y_2 = y + 1 \land g_1 > 0), \quad \overline{\psi} := (f_2 \geq 0 \land z_2 = z - 1 \land g_2 > 0).$$

where $D = \{y = \alpha(y_1), z = \alpha(z_1)\}$, $N = (y_1 = z_1 \rightarrow y = z)$.

An illustrating example

Example

$$\begin{aligned}\phi :=&(f_1 = -(y_1 - x_1 + 1)^2 - x_1 + x_2 \geq 0) \wedge (y_2 = \alpha(y_1) + 1)\\ &\wedge (g_1 = -x_1^2 - x_2^2 - y_2^2 + 1 > 0),\\ \psi :=&(f_2 = -(z_1 - x_2 + 1)^2 + x_1 - x_2 \geq 0) \wedge (z_2 = \alpha(z_1) - 1)\\ &\wedge (g_2 = -x_1^2 - x_2^2 - z_2^2 + 1 > 0).\end{aligned}$$

1. Flattening and purification gives

$$\overline{\phi} := (f_1 \geq 0 \wedge y_2 = y + 1 \wedge g_1 > 0), \quad \overline{\psi} := (f_2 \geq 0 \wedge z_2 = z - 1 \wedge g_2 > 0).$$

   where $D = \{y = \alpha(y_1), z = \alpha(z_1)\}$, $N = (y_1 = z_1 \rightarrow y = z)$.

2. **NSOSC** is not satisfied, since $h = -f_1 - f_2 = (y_1 - x_1 + 1)^2 + (z_1 - x_2 + 1)^2$ is an SOS. $h_1 = (y_1 - x_1 + 1)^2$, $h_2 = (z_1 - x_2 + 1)^2$. This gives

$$f := f_1 + h_1 = -f_2 - h_2 = -x_1 + x_2.$$

An illustrating example

3 An interpolant for ϕ, ψ is an interpolant of $((\phi \wedge f > 0) \vee (\phi \wedge f = 0))$ and $((\psi \wedge -f > 0) \vee (\phi \wedge f = 0))$ which simplifies to : $(f > 0) \vee (f \geq 0 \wedge I_2)$ where I_2 is an interpolant for $\phi \wedge f = 0$ and $\psi \wedge f = 0$. Substituting $\phi \wedge f = 0 \models y_1 = x_1 - 1$ and $\psi \wedge f = 0 \models z_1 = x_2 - 1$ into $\overline{\phi}$ and $\overline{\psi}$, we get

$$\overline{\phi'} := -x_1 + x_2 \geq 0 \wedge y_2 = y + 1 \wedge g_1 > 0 \wedge y_1 = x_1 - 1,$$

$$\overline{\psi'} := x_1 - x_2 \geq 0 \wedge z_2 = z - 1 \wedge g_2 > 0 \wedge z_1 = x_2 - 1.$$

An illustrating example

3. An interpolant for ϕ, ψ is an interpolant of $((\phi \wedge f > 0) \vee (\phi \wedge f = 0))$ and $((\psi \wedge -f > 0) \vee (\phi \wedge f = 0))$ which simplifies to : $(f > 0) \vee (f \geq 0 \wedge I_2)$ where I_2 is an interpolant for $\phi \wedge f = 0$ and $\psi \wedge f = 0$. Substituting $\phi \wedge f = 0 \models y_1 = x_1 - 1$ and $\psi \wedge f = 0 \models z_1 = x_2 - 1$ into $\overline{\phi}$ and $\overline{\psi}$, we get

$$\overline{\phi'} := -x_1 + x_2 \geq 0 \wedge y_2 = y + 1 \wedge g_1 > 0 \wedge y_1 = x_1 - 1,$$
$$\overline{\psi'} := x_1 - x_2 \geq 0 \wedge z_2 = z - 1 \wedge g_2 > 0 \wedge z_1 = x_2 - 1.$$

4. Recursively call $\mathbf{IGFQCE}$ until $\mathbf{NSOSC}$ is satisfied. $y_1 = z_1$ is deduced from linear inequalities in $\overline{\phi'}$ and $\overline{\psi'}$, and separating terms for y_1, z_1 are constructed :

$$\overline{\phi'} \models x_1 - 1 \leq y_1 \leq x_2 - 1, \quad \overline{\psi'} \models x_2 - 1 \leq z_1 \leq x_1 - 1.$$

Let $t = \alpha(x_2 - 1)$, then separate $y_1 = z_1 \rightarrow y = z$ into two parts :

$$y_1 = t^+ \rightarrow y = t, \quad t^+ = z_1 \rightarrow t = z.$$

Add them to $\overline{\phi'}$ and $\overline{\psi'}$ respectively, we have

$$\overline{\phi'}_1 := -x_1 + x_2 \geq 0 \wedge y_2 = y + 1 \wedge g_1 > 0 \wedge y_1 = x_1 - 1 \wedge y_1 = x_2 - 1 \rightarrow y = t,$$
$$\overline{\psi'}_1 := x_1 - x_2 \geq 0 \wedge z_2 = z - 1 \wedge g_2 > 0 \wedge z_1 = x_2 - 1 \wedge x_2 - 1 = z_1 \rightarrow t = z.$$

An illustrating example

4. Then

$$\overline{\phi'}_1 := -x_1 + x_2 \geq 0 \wedge y_2 = y + 1 \wedge g_1 > 0 \wedge y_1 = x_1 - 1$$
$$\wedge\, (x_2 - 1 > y_1 \vee y_1 > x_2 - 1 \vee y = t),$$
$$\overline{\psi'}_1 := x_1 - x_2 \geq 0 \wedge z_2 = z - 1 \wedge g_2 > 0 \wedge z_1 = x_2 - 1 \wedge t = z.$$

Thus,

$$\overline{\phi'}_1 := \overline{\phi'}_2 \vee \overline{\phi'}_3 \vee \overline{\phi'}_4, \textit{ where}$$
$$\overline{\phi'}_2 := -x_1 + x_2 \geq 0 \wedge y_2 = y + 1 \wedge g_1 > 0 \wedge y_1 = x_1 - 1 \wedge x_2 - 1 > y_1,$$
$$\overline{\phi'}_3 := -x_1 + x_2 \geq 0 \wedge y_2 = y + 1 \wedge g_1 > 0 \wedge y_1 = x_1 - 1 \wedge y_1 > x_2 - 1,$$
$$\overline{\phi'}_4 := -x_1 + x_2 \geq 0 \wedge y_2 = y + 1 \wedge g_1 > 0 \wedge y_1 = x_1 - 1 \wedge y = t.$$

Since $\overline{\phi'}_3 = \textit{false}$, then $\overline{\phi'}_1 = \overline{\phi'}_2 \vee \overline{\phi'}_4$. Then find interpolant $I(\overline{\phi'}_2, \overline{\psi'}_1)$ and $I(\overline{\phi'}_4, \overline{\psi'}_1)$.

An illustrating example

4. Then

$$\overline{\phi'}_1 := -x_1 + x_2 \geq 0 \wedge y_2 = y + 1 \wedge g_1 > 0 \wedge y_1 = x_1 - 1 \wedge (x_2 - 1 > y_1 \vee y_1 > x_2 - 1 \vee y = t),$$
$$\overline{\psi'}_1 := x_1 - x_2 \geq 0 \wedge z_2 = z - 1 \wedge g_2 > 0 \wedge z_1 = x_2 - 1 \wedge t = z.$$

Thus,

$$\overline{\phi'}_1 := \overline{\phi'}_2 \vee \overline{\phi'}_3 \vee \overline{\phi'}_4, \textit{ where}$$
$$\overline{\phi'}_2 := -x_1 + x_2 \geq 0 \wedge y_2 = y + 1 \wedge g_1 > 0 \wedge y_1 = x_1 - 1 \wedge x_2 - 1 > y_1,$$
$$\overline{\phi'}_3 := -x_1 + x_2 \geq 0 \wedge y_2 = y + 1 \wedge g_1 > 0 \wedge y_1 = x_1 - 1 \wedge y_1 > x_2 - 1,$$
$$\overline{\phi'}_4 := -x_1 + x_2 \geq 0 \wedge y_2 = y + 1 \wedge g_1 > 0 \wedge y_1 = x_1 - 1 \wedge y = t.$$

Since $\overline{\phi'}_3 = \textit{false}$, then $\overline{\phi'}_1 = \overline{\phi'}_2 \vee \overline{\phi'}_4$. Then find interpolant $I(\overline{\phi'}_2, \overline{\psi'}_1)$ and $I(\overline{\phi'}_4, \overline{\psi'}_1)$.

5. Finally we conclude that $I(\overline{\phi'}_2, \overline{\psi'}_1) \vee I(\overline{\phi'}_4, \overline{\psi'}_1)$ is an interpolant.

Implementation

We have implemented the presented algorithms in *Mathematica* to synthesize interpolation for concave quadratic polynomial inequalities as well as their combination with *EUF*. To deal with SOS solving and semi-definite programming, the Matlab-based optimization tool *Yalmip* and the SDP solver *SDPT3* are invoked for assistant solving.

Evaluation results

Example	Type	Time (sec)			
		CLP-Prover	Foci	CSIsat	Our Approach
Exp.1	NLA	--	--	--	0.003
Exp.2	NLA+*EUF*	--	--	--	0.036
Exp.3	NLA	--	--	--	0.014
Exp.4	NLA	--	--	--	0.003
Exp.5	LA	0.023	×	0.003	0.003
Exp.6	LA+*EUF*	0.025	0.006	0.007	0.003
Exp.7	Ellipsoid	--	--	--	0.002
Exp.8	Ellipsoid	--	--	--	0.002
Exp.9	Octagon	0.059	×	0.004	0.004
Exp.10	Octagon	0.065	×	0.004	0.004

-- means interpolant generation fails, and × specifies particularly wrong answers (satisfiable).

Related work

- McMillan [McMillan 05] popularized *interpolants* for automatically generating invariants of programs in 2005.

Related work

- McMillan [McMillan 05] popularized *interpolants* for automatically generating invariants of programs in 2005.
- Krajíček [Krajíček 97] and Pudlák [Pudlák 97] proposed approaches to deriving interpolants from *resolution proofs* prior to McMillan's work, which generate different interpolants from those done by McMillan's method.

Related work

- McMillan [McMillan 05] popularized *interpolants* for automatically generating invariants of programs in 2005.
- Krajíček [Krajíček 97] and Pudlák [Pudlák 97] proposed approaches to deriving interpolants from *resolution proofs* prior to McMillan's work, which generate different interpolants from those done by McMillan's method.
- Kapur et al. [Kapur, Majumdar & Zarba 06] established an *intimate connection* between interpolants and quantifier elimination, by which Kapur [Kapur 13] showed that interpolants form a lattice ordered using implication.

Related work

- McMillan [McMillan 05] popularized *interpolants* for automatically generating invariants of programs in 2005.
- Krajíček [Krajíček 97] and Pudlák [Pudlák 97] proposed approaches to deriving interpolants from *resolution proofs* prior to McMillan's work, which generate different interpolants from those done by McMillan's method.
- Kapur et al. [Kapur, Majumdar & Zarba 06] established an *intimate connection* between interpolants and quantifier elimination, by which Kapur [Kapur 13] showed that interpolants form a lattice ordered using implication.
- Rybalchenko et al. [Rybalchenko & Sofronie-Stokkermans 10] proposed an algorithm for generating interpolants for the combined theory of linear arithmetic and uninterpreted function symbols (*EUF*) by using a reduction of the problem to *constraint solving* in linear arithmetic.

Related work

- McMillan [McMillan 05] popularized *interpolants* for automatically generating invariants of programs in 2005.
- Krajíček [Krajíček 97] and Pudlák [Pudlák 97] proposed approaches to deriving interpolants from *resolution proofs* prior to McMillan's work, which generate different interpolants from those done by McMillan's method.
- Kapur et al. [Kapur, Majumdar & Zarba 06] established an *intimate connection* between interpolants and quantifier elimination, by which Kapur [Kapur 13] showed that interpolants form a lattice ordered using implication.
- Rybalchenko et al. [Rybalchenko & Sofronie-Stokkermans 10] proposed an algorithm for generating interpolants for the combined theory of linear arithmetic and uninterpreted function symbols (*EUF*) by using a reduction of the problem to *constraint solving* in linear arithmetic.
- Dai et al. [L. Dai, B. Xia & N. Zhan 13] provided an approach to constructing non-linear interpolants based on *semi-definite programming*.

Concluding remarks

- Contributions

Concluding remarks

- Contributions
  1. A complete, polynomial time algorithm for generating interpolants from mutually contradictory conjunctions of concave quadratic polynomial inequalities over the reals :

Concluding remarks

- Contributions
  1. A complete, polynomial time algorithm for generating interpolants from mutually contradictory conjunctions of concave quadratic polynomial inequalities over the reals :
     - If $\mathbf{NSOSC}$ holds, an interpolant a la McMillan can be generated essentially using the linearization of quadratic polynomials.

Concluding remarks

- Contributions
  1. A complete, polynomial time algorithm for generating interpolants from mutually contradictory conjunctions of concave quadratic polynomial inequalities over the reals :
     - If $\mathbf{NSOSC}$ holds, an interpolant a la McMillan can be generated essentially using the linearization of quadratic polynomials.
     - If $\mathbf{NSOSC}$ doesn't hold, linear equalities relating variables are deduced, resulting to interpolation subproblems with fewer variables on which the algorithm is recursively applied.

Concluding remarks

- Contributions
  1. A complete, polynomial time algorithm for generating interpolants from mutually contradictory conjunctions of concave quadratic polynomial inequalities over the reals :
     - If $\mathbf{NSOSC}$ holds, an interpolant a la McMillan can be generated essentially using the linearization of quadratic polynomials.
     - If $\mathbf{NSOSC}$ doesn't hold, linear equalities relating variables are deduced, resulting to interpolation subproblems with fewer variables on which the algorithm is recursively applied.
  2. An algorithm, by partitioning Horn clauses, for generating interpolants for the combination of quantifier-free theory of concave quadratic polynomial inequalities and equality theory over uninterpreted function symbols (*EUF*).

Concluding remarks

- Contributions
  1. A complete, polynomial time algorithm for generating interpolants from mutually contradictory conjunctions of concave quadratic polynomial inequalities over the reals :
     - If $\mathbf{NSOSC}$ holds, an interpolant a la McMillan can be generated essentially using the linearization of quadratic polynomials.
     - If $\mathbf{NSOSC}$ doesn't hold, linear equalities relating variables are deduced, resulting to interpolation subproblems with fewer variables on which the algorithm is recursively applied.
  2. An algorithm, by partitioning Horn clauses, for generating interpolants for the combination of quantifier-free theory of concave quadratic polynomial inequalities and equality theory over uninterpreted function symbols (*EUF*).
- Future work

Concluding remarks

- Contributions
  1. A complete, polynomial time algorithm for generating interpolants from mutually contradictory conjunctions of concave quadratic polynomial inequalities over the reals :
     - If $\mathbf{NSOSC}$ holds, an interpolant a la McMillan can be generated essentially using the linearization of quadratic polynomials.
     - If $\mathbf{NSOSC}$ doesn't hold, linear equalities relating variables are deduced, resulting to interpolation subproblems with fewer variables on which the algorithm is recursively applied.
  2. An algorithm, by partitioning Horn clauses, for generating interpolants for the combination of quantifier-free theory of concave quadratic polynomial inequalities and equality theory over uninterpreted function symbols (*EUF*).
- Future work
  - Extending the proposed framework to which their linearization with some additional conditions on the coefficients (such as concavity for quadratic polynomials).

Concluding remarks

- Contributions
  1. A complete, polynomial time algorithm for generating interpolants from mutually contradictory conjunctions of concave quadratic polynomial inequalities over the reals :
     - If $\mathbf{NSOSC}$ holds, an interpolant a la McMillan can be generated essentially using the linearization of quadratic polynomials.
     - If $\mathbf{NSOSC}$ doesn't hold, linear equalities relating variables are deduced, resulting to interpolation subproblems with fewer variables on which the algorithm is recursively applied.
  2. An algorithm, by partitioning Horn clauses, for generating interpolants for the combination of quantifier-free theory of concave quadratic polynomial inequalities and equality theory over uninterpreted function symbols (*EUF*).
- Future work
  - Extending the proposed framework to which their linearization with some additional conditions on the coefficients (such as concavity for quadratic polynomials).
  - Investigating how results reported for nonlinear polynomial inequalities based on positive nullstellensatz and the Archimedian condition on variables can be exploited in the proposed framework for dealing with polynomial inequalities.