Algebraic and Semi-Algebraic Reasoning For Formal Methods

Lecture 4 - Applications of Gröbner Bases and Sum of Squares.

Sriram Sankaranarayanan

Operations on Varieties

- Algebraic Variety V
  - Representation: Gröbner basis of the ideal $\mathrm{Id}(V)$.
- Intersection of varieties:
  - $V_1 \cap V_2$ – Groebner$(G_1 \cup G_2)$
  - Why do we need to compute Gröbner basis again?
- Union of varieties:
  - $V_1 \cup V_2$ – $G_1 \otimes G_2$ or $\langle G_1 \rangle \cap \langle G_2 \rangle$.

Unions of Varieties

Two ways of computing intersections:

$$\begin{aligned}\mathsf{Var}(\langle f_1, \dots, f_j\rangle) \cup \mathsf{Var}(\langle g_1, \dots, g_k\rangle) \\ = \mathsf{Var}(\langle f_1 g_1, f_1 g_2, \dots, f_j g_k\rangle) \\ = \mathsf{Var}(\langle f_1, \dots, f_j\rangle \cap \langle g_1, \dots, g_k\rangle)\end{aligned}$$

- How do we compute ideal intersections?
- Which one is better?

Ideal Intersection

Trick Let t be a *fresh variable*.

$$\langle f_1, \ldots, f_j \rangle \cap \langle g_1, \ldots, g_k \rangle =$$
$$\langle tf_1, \ldots, tf_j, (1-t)g_1, \ldots, (1-t)g_k \rangle \cap K[\vec{x}]$$

- Prove that this computes ideal intersection
- Eliminate the variable t.
    - We will see how to do so soon.

Ideal Products

Let $p \in \langle f_1, \ldots, f_j \rangle \cap \langle g_1, \ldots, g_k \rangle$ then $p \in \langle f_1 g_1, f_1 g_2, \ldots, f_j g_k \rangle$?

- Not necessarily!
- However, $p^2 \in \langle f_1 g_1, f_1 g_2, \ldots, f_j g_k \rangle ?$.

If $p \in \langle f_1 g_1, f_1 g_2, \ldots, f_j g_k \rangle$, is $p \in \langle f_1, \ldots, f_j \rangle \cap \langle g_1, \ldots, g_k \rangle$?

Inclusion Checking

- Algebraic Varieties: V_1, V_2.
- Check if $V_1 \subseteq V_2$.
  - Let $\langle G_1 \rangle, \langle G_2 \rangle$ be the Gröbner bases.
- If $V_1 \subseteq V_2$ then $\langle G_2 \rangle \subseteq \langle G_1 \rangle$.
- Each gen. in G_2 reduces to 0 under reduction by G_1?

Image Computation

- Image computation:
  - Assertion $\varphi : g_1(\vec{x}) = 0 \ \wedge \ \cdots \ \wedge \ g_m(\vec{x}) = 0$
  - Transition relation $\rho : \ p_1(\vec{x}, \vec{x}') = 0 \ \wedge \ \cdots \ p_m(\vec{x}, \vec{x}') = 0$.
  - Post-Condition: $(\exists \ \vec{x}_0) \ \varphi[\vec{x}_0] \wedge \ \rho[\vec{x}_0, \vec{x}]$

Elimination Theory

Let $I : \langle p_1, \ldots, p_m \rangle$ be an ideal in $K[x_1, \ldots, x_n, y_1, \ldots, y_m]$.

- Compute Gröbner basis G under an *elimination order*

Elimination Theory

Let $I : \langle p_1, \ldots, p_m \rangle$ be an ideal in $K[x_1, \ldots, x_n, y_1, \ldots, y_m]$.

- Compute Gröbner basis G under an *elimination order*
  - Eg., lexicographic ordering: $x_1 > \cdots > x_n > y_1 > \cdots > y_m$

Elimination Theory

Let $I : \langle p_1, \dots, p_m \rangle$ be an ideal in $K[x_1, \dots, x_n, y_1, \dots, y_m]$.

- Compute Gröbner basis G under an *elimination order*
  - Eg., lexicographic ordering: $x_1 > \cdots > x_n > y_1 > \cdots > y_m$
- Take all polynomials involving $y_1, \dots, y_m$:

Elimination Theory

Let $I : \langle p_1, \dots, p_m \rangle$ be an ideal in $K[x_1, \dots, x_n, y_1, \dots, y_m]$.

- Compute Gröbner basis G under an *elimination order*
  - Eg., lexicographic ordering: $x_1 > \cdots > x_n > y_1 > \cdots > y_m$
- Take all polynomials involving $y_1, \dots, y_m$:
  - $\widehat{G} = G \cap K[y_1, \dots, y_m]$

Elimination Theory

Let $I : \langle p_1, \dots, p_m \rangle$ be an ideal in $K[x_1, \dots, x_n, y_1, \dots, y_m]$.

- Compute Gröbner basis G under an *elimination order*
  - Eg., lexicographic ordering: $x_1 > \cdots > x_n > y_1 > \cdots > y_m$
- Take all polynomials involving $y_1, \dots, y_m$:
  - $\widehat{G} = G \cap K[y_1, \dots, y_m]$
- Claim: $I \cap K[y_1, \dots, y_m] = \langle \widehat{G} \rangle$.

Demonstration

Demo of using abstract interpretation to calculate polynomial invariants.

- Jupyter notebook!

Semi-Algebraic Reasoning

Lagrangian Reasoning for Inequalities

$$\begin{array}{rcll} 2x - 3y + 4z & \geq & 5 & \leftarrow e_1 \\ 3x - 2y + 0z & \geq & 7 & \leftarrow e_2 \\ z & \geq & 3 & \leftarrow e_3 \\ \hline & & (\Rightarrow) & \\ x - y + z & \geq & 3 & \end{array}$$

$$(x - y + z - 3) = \frac{1}{5}(e_1 + e_2 + e_3)$$

Lagrangian Reasoning (Continued)

$$\frac{e \geq 0, \;\; \lambda \geq 0}{\lambda e \geq 0}$$

$$\frac{e_1 \geq 0, \;\; e_2 \geq 0}{e_1 + e_2 \geq 0}$$

$$\frac{}{1 \geq 0}$$

Conic Combination

Consider linear inequalities

$$e_1 \geq 0, \dots, e_m \geq 0$$

Conic combination:

$$\lambda_0 + \lambda_1 e_1 + \cdots + \lambda_m e_m \geq 0$$

wherein $\lambda_i \geq 0$.

Farkas' Lemma

$$\varphi : \; e_1 \geq 0 \; \wedge \; e_2 \geq 0 \; \cdots \; \wedge \; e_m \geq 0$$

- φ is unsatisfiable iff $-1 \geq 0$ lies in conic combination.
- If φ is satisfiable: $\varphi \models e \geq 0$ iff we can $e = \sum_{i=1}^{m} \lambda_i e_i + \lambda_0$
    - $\lambda_0, \ldots, \lambda_m \geq 0$.
- Foundations of linear programming and duality theory.
    - Reference: V. Chvatal's amazing book on Linear Programming.

Farkas' Lemma in Formal Methods

- Vast literature on using Farkas' Lemma for
  - Invariant Synthesis: Colon + Sank. + Sipma' CAV 2003; Gulwani et al., Tiwari et al.,...
  - Ranking Function Synthesis: Colon + Sipma, Podelski + Rybalchenko, Bradley + Manna, Cook + Rybalchenko + Podelski, ...
  - Cost analysis of programs
  - Analysis of probabilistic programs
  - Proof production in linear arithmetic SMT solver: [Reynolds+Tinelli]

Beyond Farkas Lemma

Proving entailment with polynomials:

$$p_1 \geq 0, \cdots, p_m \geq 0 \models p \geq 0$$

- $p_1, \dots, p_m \in \mathbb{R}[x_1, \dots, x_n]$.
- Entailment over $\mathbb{R}^n$.

Positivstellensatz:

$$p = \sigma_0 + \sigma_1 p_1 + \cdots + \sigma_m p_m$$

where σ_i are *positive polynomials*.

Polynomial Positivity Checking

- Check if a polynomial $p \in \mathbb{R}[x_1, \ldots, x_n]$ is non-negative everywhere.

$$\forall\ x_1, \ldots, x_n \in \mathbb{R}^n,\ p(x_1, \ldots, x_n) \geq 0$$

- Well known to be co-NP hard.
  - Positive Definite: $p > 0$ for all $\vec{x} \neq 0$.
  - Positive Semi-Definite: $p \geq 0$ for all $\vec{x}$.

Semi-Algebraic Entailment Checking

Given $p_1, \ldots, p_m, p \in \mathbb{R}[x_1, \ldots, x_n]$, check entailment:

$$p_1 \geq 0 \ \wedge \ \cdots \ \wedge p_m \geq 0 \ \models \ p \geq 0 \,.$$

- Adapt Cylindrical Algebraic Decomposition.

Cylindrical Algebraic Decomposition (CAD)

Given polynomials $p_1, \dots, p_m$ in $\mathbb{R}[x_1, \dots, x_n]$,

- We decompose $\mathbb{R}^n$ into finitely many disjoint cells.
- Each cell is semi-algebraic.
- Each cell is "sign invariant".
- The decomposition is *cylindrical*.

Cylindrical Decomposition

- Decompose x_n into finitely many points and intervals.

$$\mathbb{R} = (-\infty, a_1) \cup [a_1, a_1] \cup (a_1, a_2) \cup [a_2, a_2] \cup \cdots$$

- For each point/interval, we recursively associate a "stack of regions' ' involving $x_1, \ldots, x_{n-1}$.

Cf. Manuel Kauers, How to use a Cylindrical Algebraic Decomposition, Seminaire Lotharingien de Combinatoire 65 (2011).

CAD Figure

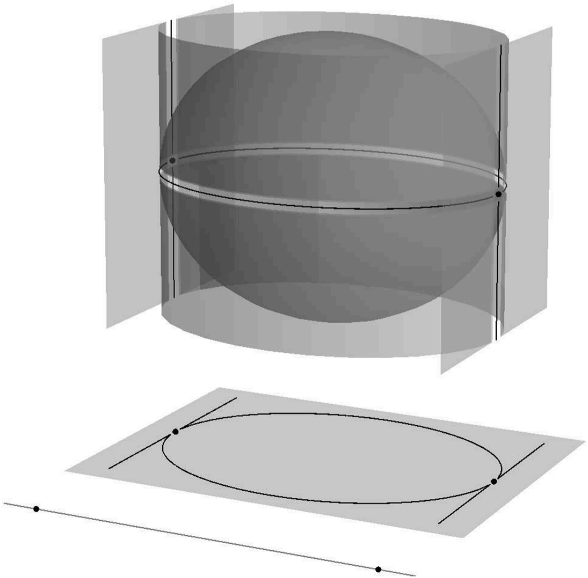

FIGURE 2. cylindrical algebraic decomposition of the unit ball

Cylindrical Algebraic Decomposition: Complexity

- CAD is a very powerful tool for working with semi-algebraic sets.
- However, its complexity is double exponential in number of variables.
  - The full CAD algorithm is not necessary to check if $\exists \vec{x}, p(\vec{x}) < 0$.

Next Session

- Sum of Squares Polynomials
- How to check if a polynomial is SOS?
  - Semi-Definite Programming.
- Positivstellensatz.