Evaluation Properties of Invariant Polynomials

Jie Wu[1,2], Eric Schost[1], Xavier Dahan[3]

[1]University of Western Ontario, Ontario, Canada
[2]Graduate School of the Chinese Academy of Sciences, China
[3]Rikkyô university, Tôkyô, Japan

May 7, 2008

Outline

- Introduction
- Main Result
- Preliminaries and Notations
- Computation of the Gröbner basis
- Complexity Analysis

Introduction

A polynomial invariant under the actions of a finite group can be rewritten as a polynomial of generators of the invariant ring by using the method of Gröbner basis.

Example

For symmetric group $\mathscr{G} = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right\}$, symmetric polynomial $f = x_1^d + x_2^d \in \mathbb{Q}[x_1, x_2]$ can be written as

$$P = \sum_{j=0}^{\lfloor d/2 \rfloor} (-1)^j \frac{d}{d-j} \binom{n-j}{j} y_1^{d-2j} y_2^j \in \mathbb{Q}[y_1, y_2]$$

with $f = P(x_1 + x_2, x_1 x_2)$ by modulo the Gröbner basis of

$$J = \langle x_1 + x_2 - y_1, x_1 x_2 - y_2 \rangle \lhd \mathbb{Q}[y_1, y_2, x_1, x_2]$$

w.r.t. lex order $x_1 > x_2 > y_1 > y_2$.

Introduction

- This rewriting method is well known in invariant theory as a classical application of Gröbner basis method.
- However, there is probably no hope to get good complexity without using a straight-line program.
- A straight-line program is a sequence of instructions $(+, -, \times)$ that computes a (sequence of) polynomial(s); the cost measure is the *size*, *i.e.*, the number of instructions. It is a useful tool to measure the complexity of evaluation properties of polynomial systems.
- It has long been known that this representation is well-adapted to obtain complexity results for questions such as multivariate factorization, GCD computation [Kaltofen] and polynomial system solving [Giusti et al.]

Who Cares?

In the problem of solving polynomial systems:

- Some algorithms benefit if the input system has a low complexity of evaluation, see
  - Giusti et al.
  - Li, Moreno Maza, Rasheed and Schost
- If the input system is symmetric
  - we want to rewrite it in polynomials of invariants
  - Without spoiling its complexity of evaluation too much

Introduction

Example

- $f = x_1^8 + x_2^8$ can be represented by a straight-line program of size 7 as: $G_1 = x_1^2$; $G_2 = G_1^2$; $G_3 = G_2^2$;$H_1 = x_2^2$; $H_2 = H_1^2$; $H_3 = H_2^2$; $f = G_3 + H_3$.
- To compute the rewritten polynomial of f by modulo by $\{x_1 + x_2 - y_1, x_2^2 - y_1 x_2 + y_2\}$, we need at most $2 \times 3 \times 11 + 3 = 69$ operations $(+, -, \times)$.
- In general, for $f = x_1^{2^k} + x_2^{2^k}$, $2 \times 11 \times k + 3$ operations are enough, here 11 is the number of operations for a multiplication in

$$\mathbb{Q}[y_1, y_2][x_1, x_2]/\langle x_1 + x_2 - y_1, x_1 x_2 - y_2 \rangle$$

Remark. One can evaluate P(the rewritten polynomial of f) within O(log(d)) arithmetic operations, comparing to the fact that P has O(d) terms.

Introduction

- In general, let $\mathscr{G}$ be a finite subgroup of $\mathrm{GL}(k, n)$, denote $\overline{x} = (x_1, \cdots, x_n)$ and $k[\overline{x}]^{\mathscr{G}} = \{f | f = \sigma \circ f, \forall \sigma \in \mathscr{G}\}$ the ring of invariants under actions of $\mathscr{G}$. Suppose that primary and minimal secondary invariants $\overline{F} = (f_1, \ldots, f_n)$ and $\overline{\sigma} = (\sigma_1, \ldots, \sigma_e)$ are known such that

$$k[\overline{x}]^{\mathscr{G}} = \bigoplus_{\sigma \in \overline{\sigma}} k[f_1, \ldots, f_n]\sigma$$

- Any $P \in k[\overline{x}]^{\mathscr{G}}$ can be uniquely written as

$$P = \sum_{\sigma \in \overline{\sigma}} P_\sigma(f_1, \ldots, f_n)\sigma$$

for some P_σ in $k[\overline{y}] = k[y_1, \ldots, y_n]$.

Introduction

Example

- $\mathscr{G} = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \right\}$
- $\overline{F} = (x_1^2, x_2^2)$ and $\overline{\sigma} = (1, x_1 x_2)$ can serve as primary and secondary invariants
- Any $P \in k[x_1, x_2]^{\mathscr{G}}$ can be uniquely written as $P = P_1(x_1^2, x_2^2) + P_{x_1 x_2}(x_1^2, x_2^2) x_1 x_2$, for $P_1, P_{x_1 x_2} \in k[y_1, y_2]$

Main Result

Theorem

Let $P \in k[\overline{x}]$ (resp. $\overline{F}, \overline{\sigma}$) be given by a straight-line program Γ of size L (resp. Γ' of size $L_{\overline{F},\overline{\sigma}}$). Given Γ and Γ', one can construct a straight-line program Γ'' of size

$$(L + L_{\overline{F},\overline{\sigma}})(n\delta)^{O(1)}$$

that computes all polynomials $(P_\sigma)_{\sigma \in \overline{\sigma}}$. Here, $\delta = \prod \deg(f_i)$

Remark. Without using a straight-line program, one probably cannot get better complexity than $\binom{n+\delta}{n}$.

General Idea

Input: $P, \overline{F}, \overline{\sigma}$

Step 1 Compute the Gröbner basis G of $J = \langle f_1 - y_1, \cdots, f_n - y_n \rangle$ w.r.t. some particular monomial order

Step 2 By following the sequence of straight-line program which represents P, compute the normal form of P modulo G

Step 3 Compute the coefficients P_σ from P mod G

Remark. If $\mathscr{G}$ is such that the only secondary invariant is 1, then Step 3 is not needed.

Preliminaries and Notations

- For simplicity, we assume $\mathscr{G}$ is a reflection group which means $k[\overline{x}]^{\mathscr{G}} = k[f_1, \cdots, f_n]$, we also assume k has characteristic 0 which is algebraically closed.
- We define the following monomial order on $k[\overline{y}, \overline{x}]$: $\overline{x}^{\alpha_1}\overline{y}^{\beta_1} > \overline{x}^{\alpha_2}\overline{y}^{\beta_2} \Leftrightarrow \overline{x}^{\alpha_1} > \overline{x}^{\alpha_2}$ for *graded lex order* $x_1 > \cdots > x_n$, or $\overline{x}^{\alpha_1} = \overline{x}^{\alpha_2}$ and $\overline{y}^{\beta_1} > \overline{y}^{\beta_2}$ for *lex order* $y_1 > \cdots > y_n$.
- Introducing degrees on y_i with degree(y_i)=degree(f_i), make J homogeneous ideal. Let $G = \{g_1(\overline{x}, \overline{y}), \cdots, g_m(\overline{x}, \overline{y})\}$ be the reduced Gröbner basis of J, thus g_i is homogeneous with weights on $y_i' s$.
- For a point $\overline{z} \in k^n$, let $\overline{p} = (p_1, \cdots, p_n) = (f_1(\overline{z}), \cdots, f_n(\overline{z}))$, $I_p = \langle f_1 - p_1, \cdots, f_n - p_n \rangle \lhd k[\overline{x}]$

Sketch of Our Algorithm

Step 1 Pick a ”lucky” point $\overline{p}$ such that I_p is radical

Step 2 Compute the Gröbner basis of I_p w.r.t graded lex order $x_1 > \cdots > x_n$

Step 3 Using a lifting technique to get G

Gröbner Basis of I_p

Theorem

For every $\overline{z} \in k^n$, $\overline{p} = (f_1(\overline{z}), \cdots, f_n(\overline{z})) \in k^n$*, the reduced Gröbner basis of* I_p *is* $\{g_1(\overline{x}, \overline{p}), \cdots, g_m(\overline{x}, \overline{p})\}$

Theorem

$\mathrm{V}(I_p) = \{s \cdot \overline{z} | s \in G\}$*,* I_p *is radical if and only if* $|\mathrm{V}(I_p)| = |\mathscr{G}|$*.*

Methods: Pick a "Lucky" $\overline{p}$, use "Shape Lemma"(interpolation) to construct a basis (powers of a "primitive element") of $k[\overline{x}]/I_p$, by using FGLM (J.C. Faugère, P. Gianni, D. Lazard, and T. Mora) algorithm, we get the Gröbner basis of I_p.

Lifting Technique

- Let $M = \langle y_1 - p_1, \cdots, y_n - p_n \rangle$ the maximal ideal in $k[\overline{y}]$, $R_i = k[\overline{y}]/M^i$

Theorem

For every i, the reduced Gröbner basis of $J \triangleleft R_i[\overline{x}]$ is $\{g \bmod M^i | g \in G\}$.

Strategy of lifting step: Let $G^k = \{g_1^k, \cdots, g_m^k\}$, where $g_i^k = g_i \bmod M^{2^k}$, be the Gröbner basis of $J \triangleleft R_{2^k}[\overline{x}]$. Suppose we know G^k, we construct a lifting step to compute G^{k+1}.

Lifting Technique

Algorithm

- *Let* $H_k = R_{2^{k+1}}[\overline{x}]/\langle G^k \rangle$*, and the image of* $\alpha \in k[\overline{y}, \overline{x}]$ *in* H_k *will be denoted by* α_k*.*
- *we treat* $F = \{f_1 - y_1, \cdots, f_n - y_n\}$ *and* G^k *as column matrices, their Jacobian matrices with respect to* $\overline{x}$ *will be denoted by* $\mathrm{Jac}(F)$ *and* $\mathrm{Jac}(G^k)$ *respectively.*
- *Compute* $\mathrm{Jac}(G^k)_k \cdot \mathrm{Jac}^{-1}(F)_k \cdot F_k$ *in* H_k*, let* δ *be its reduced form, and* $\widetilde{\delta}$ *be the preimage of* δ *in* $k[\overline{y}, \overline{x}]$*, then we have* $G^{k+1} = G^k + \widetilde{\delta}$*.*
- *If every polynomial in* G^{k+1} *is homogeneous with degree*(x_i)*=1 and degree*(y_j)*=degree*(f_j)*, we have* $G = G^{k+1}$*.*

Remark. This algorithm generates the idea of triangular lifting [E.Schost].

Complexity Estimate

Let $\delta = |\mathscr{G}|$

- Computation of Gröbner basis of I_p: O$(n\delta^3 + n^2\delta)$
- Cost of lifting:

Theorem

The cost of lifting from G^k to G^{k+1} is
$O((nL + n^3 + n^2\delta + n\delta^2)\delta^3 2^{2k+2} + (n\delta^3 + \delta^4)2^{2k+2})$,
thus the overall cost of the lifting step is
$O(((nL + n^3 + n^2\delta + n\delta^2)\delta^5 + (n\delta^3 + \delta^4)\delta^2)log(\delta))$.

Conclusion. The whole algorithm can be done in polynomial time of $n\delta$ by using straight-line program.

Thank you!