On lexicograhical Gröbner bases with special primary ideals

Dahan Xavier

Ochanomizu Univeristy, Faculty of General Educational Research

2018, Tuesday December 18th
RIMS Computer Algebra 2018 — Theorey and Applictions

My gratefullness for 12 years of support

Happy celebration Yokoyama-sensei!

Outline

Outline

Trivialities. . .

Input Pairwise coprime primary (= power of an irreducible) polynomials: $\{a_i(x)\}_{i=1,\dots,m}$.

Questions What is a generator of the ideal $I = \prod_{i=1}^{r} \langle a_i \rangle$?
What is the monomial basis $\mathrm{SM}(I)$ of $\mathbb{Q}[x]/I$?

Answer Easy: $g = \prod_i a_i(x), \quad \langle g \rangle = I$
$\mathrm{SM}(I) = \{1, x, x^2, \dots, x^{d-1}\}, \quad \deg(g) = \sum_{i=1}^{m} \deg(a_i) := d.$

Purpose How to generalize this to polynomials of several variables ?

Context of Lexicographic Gröbner bases

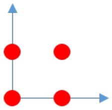
Result Complete answer when the primary ideals are triangular and verify Assumotion (**H**)(page 12)

Two variables — CRT in one variable

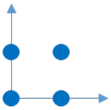
Input: three pariwise coprime primary triangular lexGbs:

$$\begin{cases} t_1^{(1)}(x) = x^2 \\ t_2^{(1)}(x,y) = y^2 + xy + 2x \end{cases} \quad \begin{cases} t_1^{(2)}(x) = x^2 \\ t_2^{(2)}(x,y) = (y+1)^2 + x(y+1) - x \end{cases}$$

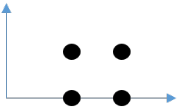
$$\begin{cases} t_1^{(3)}(x) = (x-1)^2 \\ t_2^{(3)}(x,y) = y^2 + 2(x-1)y + 3(x-1) \end{cases}$$



$\mathrm{SM}(\boldsymbol{t}^{(1)})$



$\mathrm{SM}(\boldsymbol{t}^{(2)})$



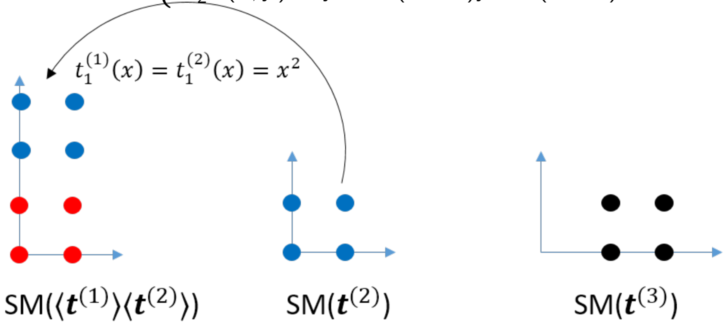
$\mathrm{SM}(\boldsymbol{t}^{(3)})$

Two variables — CRT in one variable

Input: three pariwise coprime primary triangular lexGbs:

$$\begin{cases} t_1^{(1)}(x) = x^2 \\ t_2^{(1)}(x,y) = y^2 + xy + 2x \end{cases} \qquad \begin{cases} t_1^{(2)}(x) = x^2 \\ t_2^{(2)}(x,y) = (y+1)^2 + x(y+1) - x \end{cases}$$

$$\begin{cases} t_1^{(3)}(x) = (x-1)^2 \\ t_2^{(3)}(x,y) = y^2 + 2(x-1)y + 3(x-1) \end{cases}$$

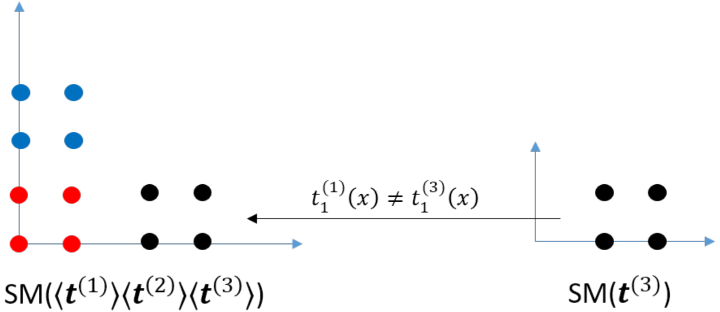


Two variables — CRT in one variable

Input: three pariwise coprime primary triangular lexGbs:

$$\begin{cases} t_1^{(1)}(x) = x^2 \\ t_2^{(1)}(x,y) = y^2 + xy + 2x \end{cases} \qquad \begin{cases} t_1^{(2)}(x) = x^2 \\ t_2^{(2)}(x,y) = (y+1)^2 + x(y+1) - x \end{cases}$$

$$\begin{cases} t_1^{(3)}(x) = (x-1)^2 \\ t_2^{(3)}(x,y) = y^2 + 2(x-1)y + 3(x-1) \end{cases}$$

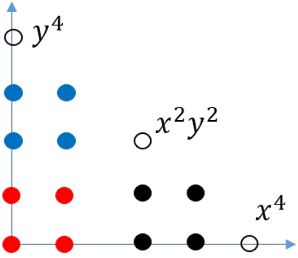


Two variables — CRT in one variable

Input: three pariwise coprime primary triangular lexGbs:

$$\begin{cases} t_1^{(1)}(x) = x^2 \\ t_2^{(1)}(x,y) = y^2 + xy + 2x \end{cases} \quad \begin{cases} t_1^{(2)}(x) = x^2 \\ t_2^{(2)}(x,y) = (y+1)^2 + x(y+1) - x \end{cases}$$

$$\begin{cases} t_1^{(3)}(x) = (x-1)^2 \\ t_2^{(3)}(x,y) = y^2 + 2(x-1)y + 3(x-1) \end{cases}$$



$\mathrm{LM}(\langle \boldsymbol{t}^{(1)} \rangle \langle \boldsymbol{t}^{(2)} \rangle \langle \boldsymbol{t}^{(3)} \rangle)$

Two variables — Previous work

Two variables is not new:

[Lazard 1985] Ideal bases and primary decomposition:case of two variables

[Gonzales-Vega, El Kahoui 1996] An improved upper complexity bound for the topology computation of a real algebraic plane curve.

[D., 2009] Size of coefficients of lexicographic Gröbner bases

[Rouillier *et al.*, 2013-2014] Computing separating linear forms for bivariate polynomials

[Schost-Mehrabi, 2015] A softly optimal monte carlo algorithm for solving bivariate polynomial systems over the integers

Two variables — Previous work

Two variables is not new:

[Lazard 1985] Ideal bases and primary decomposition:case of two variables

[Gonzales-Vega, El Kahoui 1996] An improved upper complexity bound for the topology computation of a real algebraic plane curve.

[D., 2009] Size of coefficients of lexicographic Gröbner bases

[Rouillier *et al.*, 2013-2014] Computing separating linear forms for bivariate polynomials

[Schost-Mehrabi, 2015] A softly optimal monte carlo algorithm for solving bivariate polynomial systems over the integers

Why two variables is not hard?

- managing the heap of monomials is easy
- Needs CRT (Extended GCD) in one variable only

Outline

1. introduction
2. Results
3. Idea of the methods
4. Conclusion

Results — Statement 1)-2)

Setting: $\mathcal{G}$ lexicographic Gröbner basis of a 0-dimensional ideal I

(H) All the primary ideals of I have a lexGB that is triangular.

Input: lexGB's (= triangular sets $\mathbf{t}^{(i)} = (t_1^{(i)}, \ldots, t_n^{(i)})$) of the primary components of I

(H) For all $i \neq j$, there exists a largest integer ℓ such that $t_{\leq \ell}^{(i)} = t_{\leq \ell}^{(j)}$ and $\langle t_{\ell+1}^{(i)} \rangle + \langle t_{\ell+1}^{(j)} \rangle = \langle 1 \rangle$ in $k[x_1, \ldots, x_\ell]/\langle t_{\leq \ell}^{(i)} \rangle$.

1) Standard monomials $\mathrm{SM}(I)$ can be computed with no arithmetic operations (= with no operations over k).
More precisely $O(Dnr)$ comparisons of elements in k.
r defined later, $D = |\mathrm{SM}(I)| = \dim_k(k[\mathbf{x}]/I)$ (degree of I)

2) (Chinese Remaindering Theorem – recombination) A minimal lexGB of I can be computed in $O(|\mathcal{G}| \cdot D^2)$ operations over k.
Or $O(|\mathcal{G}| \cdot D \cdot \log(D)^3)$ in the radical case (fast algorithms)

Results — Statement 3)-4)

3) Structure: let g be a polynomial in a minimal lexGB of I. There are polynomials $\chi_i \in k[x_1, \ldots, x_i]$ such that

$$\mathrm{LM}(g) = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \Rightarrow g \equiv \prod_{i=1}^{n} \chi_i \bmod \langle I_{\leq n-1} \rangle, \quad \mathrm{LM}(\chi_i) = x_i^{\alpha_i}.$$

4) Conservation of the Gröbner property under specialization maps (stability).

Rough statement: $\mathcal{G} = \{g_1, \ldots, g_s\}$. Let $\alpha = (\alpha_1, \ldots, \alpha_t) \in \bar{k}^t$ for $t < n$.
$\mathcal{G} \mid_{x_1=\alpha_1,\ldots,x_t=\alpha_t}$ still a Gröbner basis of $I \mid_{x_1=\alpha_1,\ldots,x_t=\alpha_t}$?

No in general. Yes under assumption (**H**).

What's new?

Input primary ideals are:

- Ideal of points: $\langle x_1 - a_1, \ldots, x_n - a_n \rangle$
  All results are known except the complexity of 3) (the recombination, CRT)
- Radical ideals (+ primary $\Rightarrow$ prime ideal)
  Results 3) and 4) are known.
  Results 1) and 2) are mostly new.

What's new?

Input primary ideals are:

- Ideal of points: $\langle x_1 - a_1, \dots, x_n - a_n \rangle$
  All results are known except the complexity of 3) (the recombination, CRT)
- Radical ideals (+ primary $\Rightarrow$ prime ideal)
  Results 3) and 4) are known.
  Results 1) and 2) are mostly new.
- Shifted monomial ideal
  Example: $\langle (x-1)^2, (x-1)(y+1), (y+1)^2 \rangle$.
  Results 3) and 4) have been claimed...
  but very unwieldy and checkable results
- triangular (radical or not, monomial or not) **New**

Shifted Monomial vs Triangular Primary

Fact: $\sqrt{\mathfrak{q}} := \mathfrak{p}$ has a triangular lex GB represented by polynomials:

$$(p_1(x_1)\ ,\ p_2(x_1, x_2)\ ,\ \ldots\ ,\ p_n(x_1, \ldots, x_n)),$$

where p_{i+1} is irreducible over the field $k[x_1, \ldots, x_i]/\langle p_1, \ldots, p_i \rangle$.
This encodes a "tower of field extensions".

Shifted Monomial vs Triangular Primary

Fact: $\sqrt{\mathfrak{q}} := \mathfrak{p}$ has a triangular lex GB represented by polynomials:

$$(p_1(x_1)\ ,\ p_2(x_1,x_2)\ ,\ \dots\ ,\ p_n(x_1,\dots,x_n)),$$

where p_{i+1} is irreducible over the field $k[x_1,\dots,x_i]/\langle p_1,\dots,p_i\rangle$.
This encodes a "tower of field extensions".

Proposition (Reformulation of Gianni-Trager-Zaccharias)

Any primary triangular ideal can be written as:

$$\begin{array}{rcl} T_1(x_1) & = & p_1^{e_1} \\ T_2(x_1,x_2) & = & p_2^{e_2} + \sum_{i_1=0}^{e_1-1} \sum_{i_2=0}^{e_2-1} c[i_1,i_2] p_1^{i_1} p_2^{i_2} \\ \vdots & & \\ T_n(x_1,\dots,x_n) & = & p_n^{e_n} + \sum_{i_1=0}^{e_1-1} \cdots \sum_{i_n=0}^{e_n-1} c[i_1,\dots,i_n] p_1^{i_1} \cdots p_n^{i_n} \end{array}$$

$T_\ell \equiv p_\ell^{e_\ell} \bmod \langle p_1,\dots,p_{\ell-1}\rangle \Rightarrow c[0,\dots,0,i_\ell] = 0$ *for all* i_ℓ.

Details of previous work

Work	Year	Case	Results 1) - 4)	Correctness	complexity 1) / 2)	reduced GB
This	2018	(**H**)	1) - 4)	Hopefully!	$O(rDn)$ / $O(\vert\mathcal{G}\vert.D^2)$	no
BuchMoll	1982	IdPoint	2)	$\circ$	$O(nD^3)$	yes
Abott K. Robbia.	2005	General	2)	$\circ$	$\cdot$ / $> O(nD^3)$	yes
Cerlienco Mureddu	1995	IdPoint	1) - 2)	$\circ$	$O(n^2D^2)$ / $\cdot$	no
" " "	2003	ShiftMonId	1)	$\circ$	$O(n^2D^2)/\cdot$	no
Lexgame	2006	IdPoint	1) - 2)	$\circ$	$O(rDn)/\cdot$	no
Marinari - Mora 1	2003	IdPoint	3) - 4)	Complicated	$\cdot$ / $\cdot$ (NG)	no
Maarinari - Mora 2	2006	ShiftMonId	3) - 4)	Complicated	$\cdot$ / $\cdot$ (NG)	no
Lederer	2008	IdPoint	1) - 2)	$\circ$	$\cdot$ / $\cdot$ (NG)	yes
Lei *et al*	2014	ShiftMonId	1) - 2)	Complicated	$\cdot$ / $\cdot$ (NG)	?

Result 4) Stability under specialization

Example: Consuder the lexGb for $x \prec y \prec z$.

$$\mathcal{G} = \{x^2\ ,\ y^2 + x\ ,\ xyz + y\ ,\ z^2\}.$$

$$\mathrm{LM}(\mathcal{G}) = \{x^2\ ,\ \ y^2\ ,\ \ xyx\ ,\ \ z^2\}.$$

Consider the specialization map $\phi_0\ :\ x \to 0$.

$$\phi_0(\mathrm{LM}(\mathcal{G})) = \{0\ ,\ \ y^2\ ,\ \ 0\ ,\ \ z^2\}.$$

while

$$\phi_0(\mathcal{G}) = \{0\ ,\ y^2\ ,\ y\ ,\ z^2\}.$$

Since $\mathrm{NF}(y, [y^2, z^2]) = y$ is not zero, $\phi_0(\mathcal{G})$ **is not** a lexGB.

Theorem (Stability criterion. Kalkbrener, 1997)

Let $\mathcal{G}_0 = \{g \in \mathcal{G} \mid \phi(\mathrm{LM}(\mathcal{G})) = \mathrm{LM}(\phi(\mathcal{G}))\}$.
$\mathrm{LM}(\phi(I)) = \phi(\mathrm{LM}(I)) \iff \forall g \in \mathcal{G} \setminus \mathcal{G}_0,\quad \mathrm{NF}(g, \mathcal{G}_0) = 0$

Result 4) Stability under specialization: related work

Motivation:

- Solving (Gianni - Kalkbrener)
- Parametric systems

Previous work:

[Gianni - Kalkbrener, 1987] First result in the context of specialization.

[Kalkbrener, 1997] General criterion for stability

[Becker, 1994] Prove stability for radical lexGB

Related works:

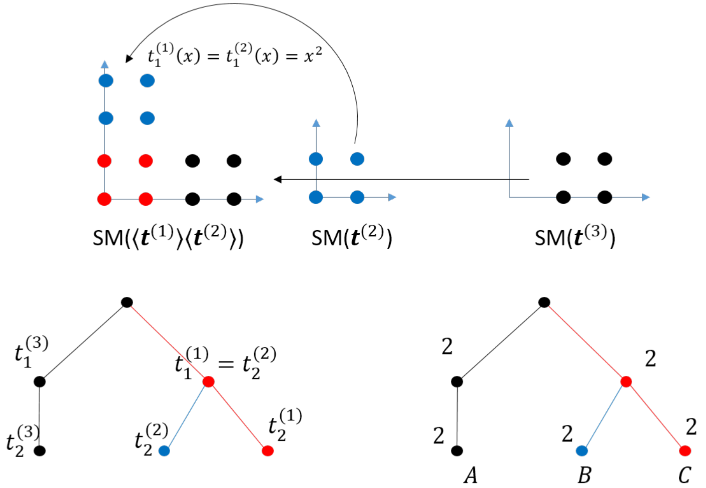
[Yokoyama, 2004, 2007], [Pan - Wang, 2006], [Weispfeinning, 2004] **Parametric exponents**

[Weispfeinning, 2003], [Kapur - Sun - Wang, 2010], [Nabeshima, 2013] Context of **Comprehensive Gröbner bases**

Outline

Result 1) - 2) Standard monomials + CRT

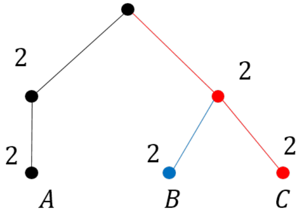
Represent the heap of monomials "cleverly": use tree data structures (following "lexgame", 2006).



Piling monomials — monomial trie

From the tree T of input lexGbs, we consruct a monomial trie U:

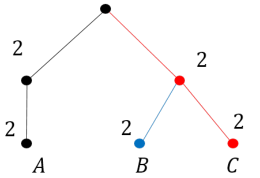
- (level 2) Leaves of T $\rightarrow$ Root of U

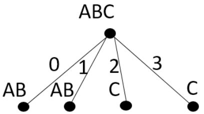


Piling monomials — monomial trie

From the tree T of input lexGbs, we consruct a monomial trie U:

- (level 1) Parent of leaves in T.
  Add the labels of the children (in T),
  record it in the labels on the edges of the trie U

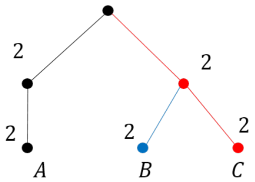


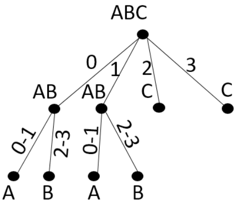


Piling monomials — monomial trie

From the tree T of input lexGbs, we consruct a monomial trie U

- (level 0) Root of T.
  Add the labels of the children of root of T in the labels on the edges of the trie U.

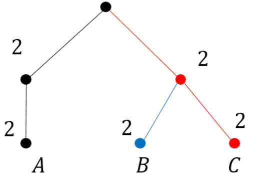


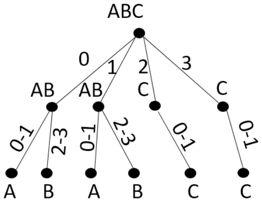


Piling monomials — monomial trie

From the tree T of input lexGbs, we consruct a monomial trie U

- (level 0) Root of T . Add the labels of the children of root of T in the labels on the edges of the trie U.





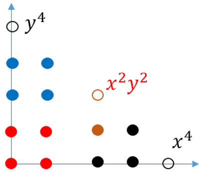
Read the standard monomials from on the edges of U from the leaves to the root of U:

$(0,0),(1,0),(2,0),(3,0),\quad(0,1),(1,1),(2,1),(3,1)$
$(0,2),(1,2)\quad(0,3),(1,3)$

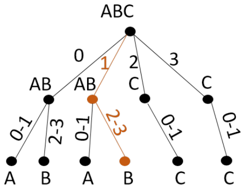
Standard monomials – Completing the proof

The proof of the algorithm above requires to construct a lexGb.
How to do?
For a polynomial involving the largest variable x_n:

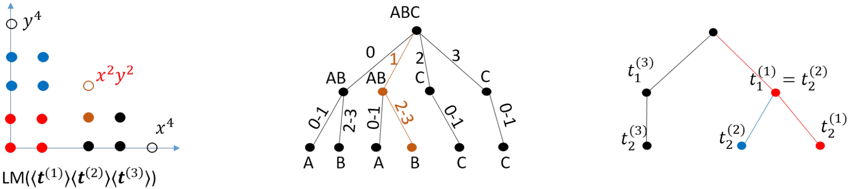
1. From $\mathrm{SM}(I)$ deduce the minimal exponents in $\mathrm{LM}(I) \cap x_n\mathrm{SM}(I)$
2. Identify the path from the leaf to the root in the trie U that contains the exponent.
3. Compute the polynomial recursively (using the tree structure).



$\mathrm{LM}(\langle \boldsymbol{t}^{(1)}\rangle\langle \boldsymbol{t}^{(2)}\rangle\langle \boldsymbol{t}^{(3)}\rangle)$

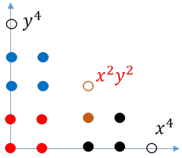


Example

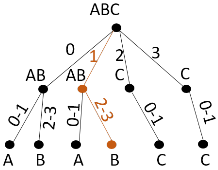


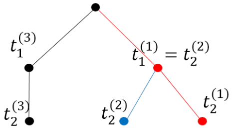
- Recursive constuction from the leaf to the root of the trie U:
- $\rightarrow$ recursive calls are made on subtrees.

Example



LM($\langle \boldsymbol{t}^{(1)}\rangle\langle \boldsymbol{t}^{(2)}\rangle\langle \boldsymbol{t}^{(3)}\rangle$)





- Recursive constuction from the leaf to the root of the trie U:
- $\rightarrow$ recursive calls are made on subtrees.
- Requires CRT to recombine output of subtrees rooted at nodes at a same level in the tree

! polynomials have coefficients modulo a primary ideal.

- CRT in defined in this context has been introduced algorithmically in:

[D., 2017] On the bit-size of non-radical triangular sets in dimension 0

- This key step is lacking in previous works.

Outline

Motivations & Applications

- Understand the structure of lexGb,
- to compute a decomposition "lexGB $\rightarrow$ triangular set" using only divisions.
- In the FGLM algorithm
  - the target order is often LEX.
  - if the lexGB is complicated this becomes heavy.
  - Can we decompose the lexGB on-the-fly to relieve the computations?

Preliminary work: (Schost - Neiger - Rakhooy. . .) 2017

Possible generalizations

- Question: Can we do the same thing for any kind of primary ideals, not only those that have a triangular lexGB?
- In theory: piling up the monomials in the "4-in-a-row" fashion should be possible.
- In general requires more sophisticated data structures than the trees introduced in the lexgame and here.
- Results 3) — Factorization pattern — and 4) — Stability under specialization – are unlikely to hold except in some special cases.

Theorem (? Reasonnable Guess)

Stability holds for $\mathcal{G}$ iff it holds for all its primary components.