Remarks on Faugère's F5 algorithm

John Perry
(based on joint work with Christian Eder)

Department of Mathematics, The University of Southern Mississippi

Sage Days 12, 21 January 2008

F5?

F5: algorithm to compute Gröbner bases of polynomial ideals

(J-C Faugère, 2002)

Outline

Gröbner bases?

Gröbner basis: "nice form" for generators of polynomial ideal

- "*nice*": ~~difficult~~ easy! questions

(B Buchberger, 1965)

Generalizes linear algebra

- *Vector space:* Gaussian elimination $\longrightarrow$ echelon form

$$\left\{\begin{array}{cccccc} * & * & * & * & = * \\ * & * & * & * & = * \\ * & * & * & * & = * \\ * & * & * & * & = * \end{array}\right. \longrightarrow \left\{\begin{array}{cccccc} * & * & * & * & = * \\ & * & * & * & = * \\ & & * & * & = * \\ & & & * & = * \end{array}\right.$$

- *Polynomial ring:* Buchberger's algorithm $\longrightarrow$ Gröbner basis

Gröbner bases?

Gröbner basis: "nice form" for generators of polynomial ideal

- "*nice*": ~~difficult~~ easy! questions

(B Buchberger, 1965)

Generalizes linear algebra

- *Vector space:* Gaussian elimination $\longrightarrow$ echelon form

$$\left\{\begin{array}{ccccc} * & * & * & * & =* \\ * & * & * & * & =* \\ * & * & * & * & =* \\ * & * & * & * & =* \end{array}\right. \longrightarrow \left\{\begin{array}{ccccc} * & * & * & * & =* \\ & * & * & * & =* \\ & & * & * & =* \\ & & & * & =* \end{array}\right.$$

- *Polynomial ring:* Buchberger's algorithm $\longrightarrow$ Gröbner basis

Buchberger's algorithm

Given $F \in \mathbb{F}\left[x_1,\ldots,x_n\right]^m$:

1. $G := F$
2. Consider all $p,q \in G$
   1. Compute $S := up - vq$ (u,p cancel $\operatorname{lcm}(\operatorname{lt}p,\operatorname{lt}q)$)
   2. Top-reduce S over G (divisibility of lts: $S - u_1g_1 - u_2g_2 - \cdots$)
   3. $S = 0? \implies$ Append S to G
3. Termination: *no new polynomials* created (Ascending Chain Condition)

- *All* GB algorithms follow this general outline (F5 too!)
- Omitting some details (lt=???)

Buchberger's algorithm

Given $F \in \mathbb{F}\left[x_1, \ldots, x_n\right]^m$:

1. $G := F$
2. Consider all $p, q \in G$
   1. Compute $S := up - vq$ (u, p cancel $\operatorname{lcm}(\operatorname{lt} p, \operatorname{lt} q)$)
   2. Top-reduce S over G (divisibility of lts: $S - u_1 g_1 - u_2 g_2 - \cdots$)
   3. $S = 0? \implies$ Append S to G
3. Termination: *no new polynomials* created (Ascending Chain Condition)

- *All* GB algorithms follow this general outline (F5 too!)
- Omitting some details (lt=???)

Buchberger's algorithm

Given $F \in \mathbb{F}\left[x_1,\ldots,x_n\right]^m$:

1. $G := F$
2. Consider all $p, q \in G$
   1. Compute $S := up - vq$ (u, p cancel $\operatorname{lcm}(\operatorname{lt} p, \operatorname{lt} q)$)
   2. Top-reduce S over G (divisibility of lts: $S - u_1 g_1 - u_2 g_2 - \cdots$)
   3. $S = 0? \implies$ Append S to G
3. Termination: *no new polynomials* created (Ascending Chain Condition)

- *All* GB algorithms follow this general outline (F5 too!)
- Omitting some details (lt=???)

Buchberger's algorithm

Given $F \in \mathbb{F}[x_1, \ldots, x_n]^m$:

1. $G := F$
2. Consider all $p, q \in G$
    1. Compute $S := up - vq$ (u, p cancel $\operatorname{lcm}(\operatorname{lt} p, \operatorname{lt} q)$)
    2. Top-reduce S over G (divisibility of lts: $S - u_1 g_1 - u_2 g_2 - \cdots$)
    3. $S = 0? \Longrightarrow$ Append S to G
3. Termination: *no new polynomials* created (Ascending Chain Condition)

- *All* GB algorithms follow this general outline (F5 too!)
- Omitting some details (lt=???)

Quick example

Problem: Find Gröbner basis of $\langle xy+1, y^2+1 \rangle$.

1. $G = (xy+1, y^2+1)$
   1. $S = y(xy+1) - x(y^2+1) = y - x$
      No top-reduction
2. $G = (xy+1, y^2+1, x-y)$
   1. $S = (xy+1) - y(x-y) = 1 + y^2$
      Top-reduces to zero
   2. $S = x(y^2+1) - y^2(x-y) = x + y^3$
      Top-reduces to zero

Quick example

Problem: Find Gröbner basis of $\langle xy+1, y^2+1 \rangle$.

1. $G = (xy+1, y^2+1)$
   1. $S = y(xy+1) - x(y^2+1) = y - x$
      No top-reduction
2. $G = (xy+1, y^2+1, x-y)$
   1. $S = (xy+1) - y(x-y) = 1 + y^2$
      Top-reduces to zero
   2. $S = x(y^2+1) - y^2(x-y) = x + y^3$
      Top-reduces to zero

Quick example

Problem: Find Gröbner basis of $\langle xy+1, y^2+1 \rangle$.

1. $G = (xy+1, y^2+1)$
   1. $S = y(xy+1) - x(y^2+1) = y - x$
      No top-reduction
2. $G = (xy+1, y^2+1, x-y)$
   1. $S = (xy+1) - y(x-y) = 1 + y^2$
      Top-reduces to zero
   2. $S = x(y^2+1) - y^2(x-y) = x + y^3$
      Top-reduces to zero

Quick example

Problem: Find Gröbner basis of $\langle xy+1, y^2+1 \rangle$.

1. $G = (xy+1, y^2+1)$
   1. $S = y(xy+1) - x(y^2+1) = y - x$
      No top-reduction
2. $G = (xy+1, y^2+1, x-y)$
   1. $S = (xy+1) - y(x-y) = 1 + y^2$
      Top-reduces to zero
   2. $S = x(y^2+1) - y^2(x-y) = x + y^3$
      Top-reduces to zero

$\therefore \mathrm{GB}(\langle xy+1, y^2+1 \rangle) = (xy+1, y^2+1, x-y)$.

Bottleneck

- Bottleneck
  - New polynomials $\rightarrow$ new information
  - Top-reduction to zero $\not\rightarrow$ no new polynomial

    $\not\rightarrow$ new information
  - $(100-\epsilon)$% of time: verifying GB, *not* computing
  - Top-reduction *very, very expensive*

Past work

- *Predict zero reductions*

  (B Buchberger 1985, R Gebauer-H Möller 1988, CKR 2002, H Hong-J Perry 2007)

- *Selection strategy:* Pick pairs in clever ways

  (B Buchberger 1985, A Giovini et al 1991, H Möller et al 1992)

- *Forbid some top-reductions:* Involutive bases

  (V Gerdt-Y Blinkov 1998)

- *Homogenization:* d-Gröbner bases

Outline

F5: overview

F5: combined approach

- Homogenize
- *d*-Gröbner bases
- New point of view:
  - New way to predict zero reductions
  - New selection strategy
- Some systems: *no* zero reductions!

"A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5)"

View from linear algebra

- Compute GB $\Longleftrightarrow$ Triangularize Sylvester matrix of G

  (D Lazard, 1983)

- Triangularize sparse matrix (F4)

  (Faugère, 1999)

- Avoid using different rows to re-compute reductions

  (Faugère, 2002)

Quick example, revisited

Problem: Find Gröbner basis of $\langle xy+1, y^2+1\rangle$.

Homogenize: $G = (xy+h^2, y^2+h^2)$

Quick example, revisited

Problem: Find Gröbner basis of $\langle xy+1, y^2+1\rangle$.
Homogenize: $G = (xy+h^2, y^2+h^2)$
$d = 2$:

No cancellations of degree 2…

Quick example, revisited

Problem: Find Gröbner basis of $\langle xy+1, y^2+1\rangle$.
Homogenize: $G = (xy+h^2, y^2+h^2)$
$d=3$:

$$\left(\begin{array}{cccccc} x^2y & xy^2 & y^3 & xh^2 & yh^2 & \\ 1 & & & 1 & & xg_1 \\ & 1 & & & 1 & yg_1 \\ & 1 & & 1 & & xg_2 \\ & & 1 & & 1 & yg_2 \end{array}\right)$$

Rows 2, 3 cancel…

Quick example, revisited

Problem: Find Gröbner basis of $\langle xy+1, y^2+1 \rangle$.

Homogenize: $G = (xy + h^2, y^2 + h^2)$

$d = 3$:

$$\begin{pmatrix} x^2y & xy^2 & y^3 & xh^2 & yh^2 & \\ 1 & & & 1 & & xg_1 \\ & 1 & & & 1 & yg_1 \\ & 1 & & 1 & & xg_2 \\ & & 1 & & 1 & yg_2 \\ & & & 1 & -1 & g_3 \end{pmatrix}$$

New! $g_3 = xh^2 - yh^2$

Quick example, revisited

Problem: Find Gröbner basis of $\langle xy+1, y^2+1\rangle$.
Homogenize: $G = (xy + h^2, y^2 + h^2)$
$d = 3$:

x^2y	xy^2	y^3	xh^2	yh^2	
1			1		xg_1
	1			1	yg_1
	~~1~~		~~1~~		~~xg_2~~ → g_3
		1		1	yg_2
			1	-1	g_3

linear dependence: ~~xg_2~~ → g_3
$(xg_2 = g_3 + yg_1)$

Quick example, revisited

Problem: Find Gröbner basis of $\langle xy+1, y^2+1\rangle$.
Homogenize: $G = (xy + h^2, y^2 + h^2)$
$d = 4$:

x^3y	x^2y^2	xy^3	y^4	x^2h^2	xyh^2	y^2h^2	h^4	
1				1				x^2g_1
	1				1			xyg_1
		1				1		y^2g_1
					1		1	h^2g_1
	~~1~~			~~1~~				~~x^2g_2~~ → xg_3
		~~1~~			~~1~~			~~xyg_2~~ → yg_3
			1			1		y^2g_2
				1	-1			xg_3
					1	-1		yg_3

linear dependence: ~~x^2g_2~~ → xg_3, ~~xyg_2~~ → yg_3

Quick example, revisited

Problem: Find Gröbner basis of $\langle xy+1, y^2+1\rangle$.
Homogenize: $G = (xy+h^2, y^2+h^2)$
$d=4$:

x^3y	x^2y^2	xy^3	y^4	x^2h^2	xyh^2	y^2h^2	h^4	
1				1				x^2g_1
	1				1			xyg_1
		1				1		y^2g_1
					1		1	h^2g_1
			1			1		y^2g_2
				1	−1			xg_3
					1	−1		yg_3

Rows 4, 7 cancel…

Quick example, revisited

Problem: Find Gröbner basis of $\langle xy+1, y^2+1\rangle$.
Homogenize: $G = \left(xy+h^2, y^2+h^2\right)$
$d=4$:

x^3y	x^2y^2	xy^3	y^4	x^2h^2	xyh^2	y^2h^2	h^4	
1				1				x^2g_1
	1				1			xyg_1
		1				1		y^2g_1
					1		1	h^2g_1
			1			1		y^2g_2
				1	-1			xg_3
					1	-1		yg_3

Rows 4, 7 cancel. . . but we will not consider them!
Why not?

Later.

Outline

Signatures

- Relation b/w rows

$$\begin{pmatrix}
x^3y & x^2y^2 & xy^3 & y^4 & x^2h^2 & xyh^2 & y^2h^2 & h^4 & \\
 & & & & & & & & \vdots \\
 & & & & & 1 & & 1 & h^2g_1 \\
 & & & & & & & & \vdots \\
 & & & & & 1 & -1 & & yg_3
\end{pmatrix}$$

and generators g_1, g_2?

- h^2g_1: obvious
- yg_3: $g_3 = xg_2 - yg_1$

Signatures

- Relation b/w rows

$$\left(\begin{array}{cccccccccc}
x^3y & x^2y^2 & xy^3 & y^4 & x^2h^2 & xyh^2 & y^2h^2 & h^4 & \\
 & & & & & & & & \vdots \\
 & & & & & 1 & & 1 & h^2g_1 \\
 & & & & & & & & \vdots \\
 & & & & & 1 & -1 & & yg_3
\end{array}\right)$$

and generators g_1, g_2?

- $h^2 g_1$: obvious
- yg_3: $g_3 = xg_2 - yg_1$

Signature of g_3: $\text{Sig}(g_3) = xg_2$.

$\therefore\ \text{Sig}(yg_3) = xyg_2$.

Signatures: Observations

- $\text{Sig}(p) = tg_i$?
  - $1 \le i \le m$ (inputs: $(g_1, \ldots, g_m)$)
  - $g = h_1 g_1 + \cdots + h_{i-1} g_{i-1} + (t + \cdots) g_i$ $(\exists h_1, \ldots, h_i,\ \text{lt}(h_i) = t)$
- this definition = algorithmic behavior
  ≠ Faugère's definition
- "easy" record-keeping: list of rules
- "easily" reject certain useless pairs:
  - Use yg_3 w/sig xyg_2, not xyg_2
  - Use xg_3 w/sig x^2g_2, not x^2g_2
  - …
- Criterion "Rewritten"

(J-C Faugère 2007?, J Gash 2008, C Eder-J Perry submitted)

Signatures: Observations

- $\text{Sig}(p) = tg_i$?
  - $1 \leq i \leq m$ (inputs: $(g_1, \ldots, g_m)$)
  - $g = h_1 g_1 + \cdots + h_{i-1} g_{i-1} + (t + \cdots) g_i$ $(\exists h_1, \ldots, h_i, \text{lt}(h_i) = t)$
- this definition = algorithmic behavior
  $\neq$ Faugère's definition
- "easy" record-keeping: list of rules
- "easily" reject certain useless pairs:
  - Use yg_3 w/sig xyg_2, not xyg_2
  - Use xg_3 w/sig $x^2 g_2$, not $x^2 g_2$
  - …
- Criterion "Rewritten"

(J-C Faugère 2007?, J Gash 2008, C Eder-J Perry submitted)

Faugère's characterization

Theorem (Faugère, 2002)

$$(A) \iff (B) \text{ where}$$

(A) G a Gröbner basis
(B) $\forall p, q \in G$ *where*

- $u\mathrm{Sig}(p)$, $v\mathrm{Sig}(q)$ *not rewritable, and*
- $u\mathrm{Sig}(p)$, $v\mathrm{Sig}(q)$ *minimal*

S-polynomial $up - vq$ *top-reduces to zero w/out changing signature*

(highly paraphrased, slightly generalized)

Outline

How to predict zero reductions?

- Recall

$$\begin{pmatrix} x^3y & x^2y^2 & xy^3 & y^4 & x^2h^2 & xyh^2 & y^2h^2 & h^4 & \\ & & & & & & & & \vdots \\ & & & & & 1 & & -1 & h^2g_1 \\ & & & & & & & & \vdots \\ & & & & & 1 & -1 & & yg_3 \end{pmatrix}$$

We did not cancel. *Why not?*

- *S*-poly top-reduces to zero
- *can predict this*

How?

How to predict zero reductions?

- Recall

$$\left(\begin{array}{cccccccccc} x^3y & x^2y^2 & xy^3 & y^4 & x^2h^2 & xyh^2 & y^2h^2 & h^4 & \\ & & & & & & & & \vdots \\ & & & & & 1 & & -1 & h^2g_1 \\ & & & & & & & & \vdots \\ & & & & & 1 & -1 & & yg_3 \end{array}\right)$$

We did not cancel. *Why not?*

- S-poly top-reduces to zero
- *can predict this*

How?

Faugère's criterion

Theorem

If

- $u\mathrm{Sig}(p) = ug_i$; *and*
- $\mathrm{lt}(q) \mid u$, $\exists q \in \mathrm{GB}_{\mathrm{prev}}(g_1, \ldots, g_{i-1})$;

then $u\mathrm{Sig}(p)$ *is not minimal.*

Definition

$\mathrm{FC}(u\mathrm{Sig}(p))$: $\quad \mathrm{lt}(q) \mid u \ \exists q \in G_{\mathrm{prev}}$

Corollary

In S-polynomial $up - vq$,

if $FC(u\mathrm{Sig}(p))$ *or* $FC(v\mathrm{Sig}(q))$

then we need not compute S.

Faugère's criterion

Theorem

If

- $u\mathrm{Sig}(p) = ug_i$; *and*
- $\mathrm{lt}(q) \mid u$, $\exists q \in \mathrm{GB}_{\mathrm{prev}}(g_1, \ldots, g_{i-1})$;

then $u\mathrm{Sig}(p)$ *is not minimal.*

Definition

$\mathrm{FC}(u\mathrm{Sig}(p))$: $\quad \mathrm{lt}(q) \mid u \ \exists q \in \mathrm{G}_{\mathrm{prev}}$

Corollary

In S-polynomial $up - vq$,

if $FC(u\mathrm{Sig}(p))$ *or* $FC(v\mathrm{Sig}(q))$

then we need not compute S.

In the example…

- Recall

$$\begin{pmatrix} x^3y & x^2y^2 & xy^3 & y^4 & x^2h^2 & xyh^2 & y^2h^2 & h^4 & \\ & & & & & & & & \vdots \\ & & & & & 1 & & -1 & h^2g_1 \\ & & & & & & & & \vdots \\ & & & & & 1 & -1 & & yg_3 \end{pmatrix}$$

- $G_{\text{prev}} = (g_1)$
- $\text{Sig}(g_3) = xg_2$
- $y\text{Sig}(g_3) = xyg_2$, and $\text{lt}(g_1) \mid xy$…

FC $\Longrightarrow$ no need to compute S-polynomial

Why?

In the example…

- Recall

$$\begin{pmatrix} x^3y & x^2y^2 & xy^3 & y^4 & x^2h^2 & xyh^2 & y^2h^2 & h^4 & \\ & & & & & & & & \vdots \\ & & & & & 1 & & -1 & h^2g_1 \\ & & & & & & & & \vdots \\ & & & & & 1 & -1 & & yg_3 \end{pmatrix}$$

- $G_{\text{prev}} = (g_1)$
- $\text{Sig}(g_3) = xg_2$
- $y\text{Sig}(g_3) = xyg_2$, and $\text{lt}(g_1) \mid xy\ldots$

FC $\Longrightarrow$ no need to compute S-polynomial

Why?

Why? Trivial syzygies

Recall $yg_3 = y[xg_2 - yg_1]\ldots$

Why? Trivial syzygies

Recall $yg_3 = y[xg_2 - yg_1]\ldots$

$$\begin{aligned} \therefore yg_3 &= y[xg_2 - yg_1] \\ &= xyg_2 - y^2 g_1 \end{aligned}$$

Why? Trivial syzygies

Recall $yg_3 = y\left[xg_2 - yg_1\right]\ldots$

$$\therefore yg_3 = y\left[xg_2 - yg_1\right]$$
$$= xyg_2 - y^2g_1$$

Trivially $g_1g_2 - g_2g_1 = 0$.

Why? Trivial syzygies

Recall $yg_3 = y[xg_2 - yg_1]\ldots$

$$\therefore yg_3 = y[xg_2 - yg_1]$$
$$= xyg_2 - y^2g_1$$

Trivially $g_1g_2 - g_2g_1 = 0$.

$$\therefore yg_3 = xyg_2 - y^2g_1 - \left[\left(xy + h^2\right)g_2 - \left(y^2 + h^2\right)g_1\right]$$
$$= -h^2g_2 + h^2g_1$$

Why? Trivial syzygies

Recall $yg_3 = y[xg_2 - yg_1]\ldots$

$$\begin{aligned}\therefore yg_3 &= y[xg_2 - yg_1] \\ &= xyg_2 - y^2 g_1\end{aligned}$$

Trivially $g_1 g_2 - g_2 g_1 = 0$.

$$\begin{aligned}\therefore yg_3 &= xyg_2 - y^2 g_1 \\ &\quad - \left[\left(xy + h^2\right) g_2 - \left(y^2 + h^2\right) g_1\right] \\ &= -h^2 g_2 + h^2 g_1\end{aligned}$$

$\mathrm{Sig}(yg_3)$ not minimal!

Outline

The F5 Algorithm

1. Each stage: Incremental strategy
   1. Compute GB (g_1)
   2. Compute GB (g_1, g_2)
   3. …
2. d-GB's $\rightsquigarrow$ GB $(g_1, \ldots, g_i)$
3. only S-polys with
   - signatures that do not satisfy (FC); *and*
   - non-rewritable components.
4. Top-reduce, but not if reduction…
   1. satisfies (FC); *or*
   2. rewritable.
5. Track new polys with signature

The F5 Algorithm

1. Each stage: Incremental strategy
    1. Compute $\mathrm{GB}(g_1)$
    2. Compute $\mathrm{GB}(g_1, g_2)$
    3. ...
2. d-GB's $\rightsquigarrow$ $\mathrm{GB}(g_1, \ldots, g_i)$
3. only S-polys with
    - signatures that do not satisfy (FC); *and*
    - non-rewritable components.
4. Top-reduce, but not if reduction…
    1. satisfies (FC); *or*
    2. rewritable.
5. Track new polys with signature

The F5 Algorithm

1. Each stage: Incremental strategy
   1. Compute GB (g_1)
   2. Compute GB (g_1, g_2)
   3. ...
2. d-GB's $\rightsquigarrow$ GB $(g_1, \ldots, g_i)$
3. only S-polys with
   - signatures that do not satisfy (FC); *and*
   - non-rewritable components.
4. Top-reduce, but not if reduction...
   1. satisfies (FC); *or*
   2. rewritable.
5. Track new polys with signature

Certain details omitted...

Zero reductions?

Definition

If $G = (g_1, \ldots, g_m)$ has trivial syzygies *only*, then G is a **regular sequence**.

> *Many systems are regular sequences;*
> *many non-regular systems can be rewritten as regular.*

Corollary

If input to F5 is a regular sequence, then no zero reductions occur.

Zero reductions?

Definition

If $G = (g_1, \ldots, g_m)$ has trivial syzygies *only*, then G is a **regular sequence**.

Many systems are regular sequences;
many non-regular systems can be rewritten as regular.

Corollary

If input to F5 is a regular sequence, then no zero reductions occur.

Relation to Buchberger's criteria?

None.

- F5 needs to compute signatures
- Buchberger's criteria ignorant of signatures
- Mixing leads to non-termination
- (but see Gash, 2008)

Relation to Buchberger's criteria?

None.

- F5 needs to compute signatures
- Buchberger's criteria ignorant of signatures
- Mixing leads to non-termination
- (but see Gash, 2008)

Outline

Motivation

- little public code...
  - Stegers: Magma
  - I don't have Magma
  - I like Sage, can use Maple
  - FGb source code not public
- compare with other algorithms
  - selection strategy
  - predicting zero reduction
  - time/space tradeoff?

Outline

Implementations (1)

- Faugère (2002)
  - C, interfaces w/Maple
  - *Very* fast
  - Several variants: F5, F5', F5", ...?
  - Souce code not publicly available, binary download
- Stegers (2005)
  - Interpreted Magma code
  - Respectable timings
  - Variant "F5R"
  - `http://wwwcsif.cs.ucdavis.edu/~stegers/`
- Others
  - Unstable implementations
  - Magma implementation?

Implementations (2)

- Perry (2007)
  - Interpreted Maple code
  - Embarassingly slow
  - Source code ~~publicly available~~ unmaintained
- Eder, Perry (2008)
  - Interpreted Singular code
  - Respectable timings
  - New variant "F5C"
  - `http://www.math.usm.edu/perry/research.html`

Implementations (3)

- Albrecht (2008)
  - Interpreted Sage/Python code
  - Faster than Eder, Perry (2008)
  - Variants F5, F5R, F5C
  - `http://bitbucket.org/malb/algebraic_attacks/`
- King (2008)
  - Compiled Sage/Cython code
  - Faster than Eder, Perry (2008) and Albrecht (2008)?
  - Variant F5R; variants F5 and F5C by Perry
  - `http://www.math.usm.edu/perry/research.html`
- Eder (in progress)
  - *F5 in Singular kernel*
  - Access to many Singular optimizations
  - Sage uses Singular, so direct benefit to Sage
  - Source code will be publicly available

So you want to implement F5…

- Faugère's pseudocode:

  `www-spaces.lip6.fr/@papers/F02a.pdf`

  (2004 edition, corrected!)

- Stegers' pseudocode:

  `wwwcsif.cs.ucdavis.edu/~stegers/`

  (contains errors)

- Perry's pseudocode:

  `www.math.usm.edu/perry/research.html`

  (used for Singular, Sage implementations)

Outline

Reduced Gröbner basis

- Some inefficiency in F5
  - Not all top-reductions allowed
  - Redundant lt's added
  - Necessary this stage, but...
  - ...*not* next stages, *not* for GB
- *Reduced* Gröbner basis?
  - Pruning of redundant lt's
  - Well-known optimization
- "Naïve" F5 does not use RGB

Reduced Gröbner basis

- Some inefficiency in F5
  - Not all top-reductions allowed
  - Redundant lt's added
  - Necessary this stage, but…
  - …*not* next stages, *not* for GB
- *Reduced* Gröbner basis?
  - Pruning of redundant lt's
  - Well-known optimization
- "Naïve" F5 does not use RGB

F5R (Stegers, 2006)

- Compute GB G of $\langle f_1, \ldots f_i \rangle$ (usual F5)
- Compute RGB B of $\langle G \rangle$ (easy: interreduce G)
- Compute GB of $\langle f_1, \ldots, f_{i+1} \rangle$
  - Use G for critical pairs, B for top-reduction
- *Many* fewer reductions than F5, but…
- Same # polys considered, generated

F5C (Eder and Perry, 2008–2009)

- Compute GB G of $\langle f_1, \ldots, f_i \rangle$ (usual F5)
- Compute RGB B of $\langle G \rangle$ (usual F5R)
- Compute GB of $\langle f_1, \ldots, f_{i+1} \rangle$
  - Use B for top-reduction *and* for critical pairs
  - Modify rewrite rules
- Significantly fewer reductions than F5R, and…
- Fewer polys considered, generated

#Critical pairs, #Polynomials in variants

F5, F5R			F5C		
i	$\#G_{\text{curr}}$	$\max\{\#P_d\}$	i	$\#G_{\text{curr}}$	$\max\{\#P_d\}$
2	2	N/A	2	2	N/A
3	4	1	3	4	1
4	8	2	4	8	2
5	16	4	5	15	4
6	32	8	6	29	6
7	60	17	7	51	12
8	132	29	8	109	29
9	524	89	9	472	71
10	1165	276	10	778	89

#Reductions

variant:	F5	F5R	F5C
Katsura-5	346	289	222
Katsura-6	8,357	2,107	1,383
Katsura-7	1,025,408	24,719	10,000
Cyclic-5	441	457	415
Cyclic-6	36,139	17,512	10,970

(Top-reduction, normal forms)

(*Many* more in Gebauer-Möller: $> 1,500,000$ in Cyclic-6)

Outline

Termination: the difficulty

Termination?

- Buchberger: ACC $\Longrightarrow$ S-polys reduce to zero eventually
- Faugère: S-polys w/minimal signatures computed, *but*. . .

Termination: the difficulty

Termination?

- Buchberger: ACC $\Longrightarrow$ S-polys reduce to zero eventually
- Faugère: S-polys w/minimal signatures computed, *but*…
  - Some top-reductions forbidden
  - Regular system: no zero reductions
  - How recognize GB property?

Outline

Faugère's original argument

Theorem

If reduction stage concludes without zero reductions, then ideal of lt's has increased.

Example

S-polynomial of $f_1 = xy + 1$, $f_2 = y^2 + 1$ did not reduce to zero;
new polynomial $x - y$;
new lt x!

Faugère's original argument

Theorem

If reduction stage concludes without zero reductions, then ideal of lt's has increased.

This theorem is wrong.

Example (Gash, 2008)

- Uses Faugère's example (2002 paper)
- Consider S-polynomials in different order
- $\rightsquigarrow$ no reduction to zero
  and ideal of lt's does not increase.
- **"redundant polynomials"**

Redundant polynomials: necessary?

Why does F5 compute redundant polynomials?

- Some top-reductions forbidden
- Redundant polynomials restore necessary top-reductions

Example

- p_1 top-reducible by p_2, but forbidden
- p_1 added to GB $\rightsquigarrow$ new rewrite rule
- p_3 top-reducible by p_1? *now allowed*
- equivalent to top-reduction by p_2

Redundant polynomials: necessary?

Why does F5 compute redundant polynomials?

- Some top-reductions forbidden
- Redundant polynomials restore necessary top-reductions

Example

- p_1 top-reducible by p_2, but forbidden
- p_1 added to GB $\rightsquigarrow$ new rewrite rule
- p_3 top-reducible by p_1? *now allowed*
- equivalent to top-reduction by p_2

Possible resolution. . . ?

An idea:

- Suppose reduction stage returns redundant polynomials
    - d-Gröbner basis!
- keep polys, but. . .
- not their S-polys
    - multiples of reducers' S-polynomials
- **Guaranteed termination!** *but*. . .
- No longer guaranteed correct!
    - Non-trivial concern: Cyclic-7 oops!
    - Rewrite rules $\Longrightarrow$ non-computed S-polys!

Possible resolution. . . ?

An idea:

- Suppose reduction stage returns redundant polynomials
    - d-Gröbner basis!
- keep polys, but. . .
- not their S-polys
    - multiples of reducers' S-polynomials
- **Guaranteed termination!** *but*. . .
- No longer guaranteed correct!
    - Non-trivial concern: Cyclic-7 oops!
    - Rewrite rules $\Longrightarrow$ non-computed S-polys!

Regular case

- General agreement: termination
- Proof in Faugère's HDR? (2007)
- Another idea (J Gash, 2009)
  - Non-termination? chain of divisible lt's
  - Subchain of divisible signatures (ACC)
  - Cannot occur in regular case
  - Still working on this…

Outline

Non-terminating examples

- Widespread belief: F5 does not always terminate
- Proposals for non-terminating systems
  - Stegers' `nonTerminatingExample.mag`
  - Brickenstein's example
    (private communication, exploit iterative computation)
- However...
  - Singular and Sage: *both* systems terminate

nonTerminatingExample.mag

Termination in Singular and Sage, not in Magma?!?

- Error in implementation
  - Rewrite rules sometimes not assigned
  - Some top-reductions not completed
- Correction $\rightsquigarrow$ termination!

(R Dellaca-J Gash-J Perry, 2009)

Outline

Private communications

- Faugère, 2007 HDR: proof fixed
  - Regular sequences only?
  - Find me a copy?
- Zobnin, 2008: Restructured algorithm
  - Proceeds by increasing signature, other changes
  - Implementation?

F5t

Gash (2008 PhD Dissertation)

- Redundant polynomials $\rightsquigarrow$ special bin D
- Test for GB: force carefully-chosen zero reductions
- If failure, add D to GB and proceed
- Loss of efficiency via zero reductions vs. guaranteed termination and correctness

Another solution?

Another idea: modified F5C

- Suppose reduction stage returns redundant polynomials
  - d-Gröbner basis!
- Immediately interreduce, discard *all* redundant polynomials
- Re-examine all pairs
  - S-polynomials of degree $\leq d$: good! new rewrite rule
  - S-polynomials of degree $> d$: bad! compute S-poly
- **WARNING:**

  The above has not (yet) been proved or implemented.

Finis

Thank you!