Understanding and Implementing F5

John Perry

`john.perry@usm.edu`

University of Southern Mississippi

Overview

Understanding F5

- Description
- Criteria
- Proofs

Implementing F5

- Maple, Singular
- Relation to Buchberger's Criteria
- Some optimizations

Motivation

- challenge!
- compare with other criteria
  - Buchberger criteria
  - Pivoting, Extended First Criteria
- open source / something to work with

Part 1

Understanding F5

Introduction

BA (1965) GB algorithm

GM (1988) fast GB algorithm

- selection strategy: normal
- near-optimal use of Buchberger's Criteria

Introduction

F4 (1999) *FAST* GB algorithm

- selection strategy: increasing total degree
- sparse linear algebra

F5 (2002) ***FAAAAAAAST*** GB algorithm (sometimes*)

- incremental selection strategy
  - $(f_1), (f_1, f_2), \ldots, (f_1, f_2, \ldots, f_m)$
- new criteria
  - Buchberger criteria disregarded
- regular sequence $\Longrightarrow$ *no reduction to zero*

*“sometimes”: If close to GB, and most real-world systems in engineering are close to GB. (Faugère)

Introduction

F4 (1999) *FAST* GB algorithm

- well understood
- many implementations, widely available

F5 (2002) ***FAAAAAAAST*** GB algorithm (sometimes)

- not well understood
- few implementations (5)*
- fewer work

*Stegers' count, (mid-2005)

Introduction

2002 paper "limited length" ⇝ algorithm vs. proofs?

- full description of algorithm
- sketch of proofs

2002—2007 problems solved w/F5

- no new description of algorithm
- no proofs

2007 HDR ? not in wide circulation

Idea

Linear system:

$$f_1 = 2x + 3y - 8$$
$$f_2 = 4x + 5y - 12$$

Idea

Linear system:

$$\begin{pmatrix} 2 & 3 & -8 \\ 4 & 5 & -12 \end{pmatrix}$$

Idea

Linear system:

$$\begin{pmatrix} 2 & 3 & -8 \\ 4 & 5 & -12 \\ & 1 & -4 \end{pmatrix}$$

$$(f_3 = 2f_1 - f_2)$$

Idea

Linear system:

$$\begin{pmatrix} 2 & 3 & -8 \\ 4 & 5 & -12 \\ & 1 & -4 \end{pmatrix}$$

Moral:

- $f_3 = 2f_1 - f_2 \rightsquigarrow$ discard f_1 or f_2 from matrix
- linear dependence $\rightsquigarrow$ f_2 is *rewritable* by f_3
- R criterion

Linear to non-linear

Compute GB $\Leftrightarrow$ Triangularize Sylvester submatrix

(Lazard, 1983)

- S-poly $\longleftrightarrow$ two rows needing cancellation
- $p \xrightarrow[G]{*} 0$ iff linear dependence in Syl(G)

Linear to non-linear

Example: $f_1 = x^2 - 1,\ f_2 = xy - 1$

$$
\begin{pmatrix}
x^3 & x^2y & xy^2 & y^3 & x^2 & xy & y^2 & x & y & 1 & \\
1 & & & & & & & -1 & & & xf_1 \\
 & 1 & & & & & & & -1 & & yf_1 \\
 & 1 & & & & & & -1 & & & xf_2 \\
 & & 1 & & & & & & -1 & & yf_2 \\
 & & & & 1 & & & & & -1 & f_1 \\
 & & & & & 1 & & & & -1 & f_2
\end{pmatrix}
$$

Linear to non-linear

Example: $f_1 = x^2 - 1,\ f_2 = xy - 1$

x^3	x^2y	xy^2	y^3	x^2	xy	y^2	x	y	1	
1							−1			xf_1
	1							−1		yf_1
							1	−1		~~xf_2~~ f_3
		1						−1		yf_2
				1					−1	f_1
					1				−1	f_2

$$f_3 = yf_1 - xf_2$$

Linear to non-linear

Example: $f_1 = x^2 - 1,\ f_2 = xy - 1$

$$\left(\begin{array}{cccccccccc|c}
x^3 & x^2y & xy^2 & y^3 & x^2 & xy & y^2 & x & y & 1 & \\
1 & & & & & & & -1 & & & xf_1 \\
 & 1 & & & & & & & -1 & & yf_1 \\
 & & 1 & & & & & & -1 & & yf_2 \\
 & & & & 1 & & & & & -1 & f_1 \\
 & & & & 1 & -1 & & & & & xf_3 \\
 & & & & & 1 & & & & -1 & f_2 \\
 & & & & & 1 & -1 & & & & yf_3 \\
 & & & & & & & 1 & -1 & & f_3
\end{array}\right)$$

Linear to non-linear

Example: $f_1 = x^2 - 1$, $f_2 = xy - 1$

x^3	x^2y	xy^2	y^3	x^2	xy	y^2	x	y	1	
1							−1			$x f_1$
	1							−1		$y f_1$
		1						−1		$y f_2$
				1					−1	f_1
				1	−1					$x f_3$
					1				−1	f_2
						1			−1	~~$y f_3$~~ f_4
							1	−1		f_3

$$f_4 = f_2 - y f_3$$

Linear to non-linear

Example: $f_1 = x^2 - 1$, $f_2 = xy - 1$

$$
\begin{pmatrix}
x^3 & x^2y & xy^2 & y^3 & x^2 & xy & y^2 & x & y & 1 & \\
1 & & & & & & & -1 & & & xf_1 \\
 & 1 & & & & & & & -1 & & yf_1 \\
 & & 1 & & & & & & -1 & & yf_2 \\
 & & & & 1 & & & & & -1 & f_1 \\
 & & & & 1 & -1 & & & & & xf_3 \\
 & & & & & 1 & & & & -1 & f_2 \\
 & & & & & & 1 & & & -1 & f_4 \\
 & & & & & & & 1 & -1 & & f_3
\end{pmatrix}
$$

Linear to non-linear

Example: $f_1 = x^2 - 1$, $f_2 = xy - 1$

x^3	x^2y	xy^2	y^3	x^2	xy	y^2	x	y	1	
1							−1			xf_1
	1							−1		yf_1
		1						−1		yf_2
				1					−1	f_1
					1				−1	~~xf_3~~ f_5
					1				−1	f_2
						1			−1	f_4
							1	−1		f_3

$$f_5 = f_1 - xf_3$$

*Note $f_5 = f_2 \rightsquigarrow$ no new info $\rightsquigarrow f_1 - xf_3$ “useless”

Linear to non-linear

Example: $f_1 = x^2 - 1$, $f_2 = xy - 1$

$$\left(\begin{array}{ccccccccccc}
x^3 & x^2y & xy^2 & y^3 & x^2 & xy & y^2 & x & y & 1 & \\
1 & & & & & & & -1 & & & xf_1 \\
 & 1 & & & & & & & -1 & & yf_1 \\
 & & 1 & & & & & & -1 & & yf_2 \\
 & & & & 1 & & & & & -1 & f_1 \\
 & & & & & 1 & & & & -1 & f_2 \\
 & & & & & & 1 & & & -1 & f_4 \\
 & & & & & & & 1 & -1 & & f_3
\end{array}\right)$$

Higher order cancellations also useless.

Basis: $G = \{f_1, f_2, f_3, f_4\} \rightsquigarrow$ Reduced basis: $R = \{f_3, f_4\}$

Idea

Signatures: *multiplier* and *index*

$$
\begin{aligned}
f_3 &= y f_1 - x f_2 \\
f_4 &= f_1 - x f_3 \\
&= f_1 - x\,(y f_1 - x f_2) \\
&= (-xy + 1)\,f_1 + x^2 f_2 \\
f_5 &= f_2 - y f_3 \\
&= f_2 - y\,(y f_1 - x f_2) \\
&= -y^2 f_1 + (xy + 1)\,f_2
\end{aligned}
$$

Idea

Signature

- "leading term" of representation
- rewritable ⇝ linear dependence
- remainders (f_3, f_4, f_5) *not rewritable*
- *respect the signatures!!!*
  - consider all cancellations in Syl
  - avoid cancellations that might affect signature
  - top-reduction: new polynomial?

Summary of idea

- compute GB $\Longleftrightarrow$ triangularize Syl
- R criterion: older rows $\rightsquigarrow$ newer rows
- signatures: track linear dependencies

Basic description of algorithm

- compute GB of $(f_1, f_2) \rightsquigarrow G_2 = \left(g_1, g_2, \ldots, g_{m_2}\right)$
- compute GB of (f_1, f_2, f_3)
  - same as GB of $\left(g_1, g_2, \ldots, g_{m_2}, f_3\right)$
    $\rightsquigarrow G_3 = \left(g_1, g_2, \ldots, g_{m_3}\right)$
- **iterate:** after computing G_i, compute GB of $\left(g_1, g_2, \ldots, g_{m_i}, f_{i+1}\right)$ $1 < i < m$
- Use signatures to avoid linear dependencies / useless cancellations

New criterion

Question: Could we detect $f_1 - xf_3 \xrightarrow[F]{*} 0$?

Answer: Principal Syzygies (PS)

$$f_1 f_2 - f_2 f_1 = 0$$

Sums of monomial multiples of f_1, f_2

$\rightsquigarrow$ linear dependence in Sylvester matrix

New criterion

Example:

$$f_1 - x f_3 = (-xy+1) f_1 + x^2 f_2.$$

But

$$f_1 f_2 - f_2 f_1 = 0 \Longrightarrow \left(x^2 - 1\right) f_2 - (xy-1) f_1 = 0$$
$$\Longrightarrow x^2 f_2 = (xy-1) f_1 + f_2.$$

Thus

$$f_1 - x f_3 = (-xy+1) f_1 + [(xy-1) f_1 + f_2] = 1 f_2.$$

Lowered signature: already considered any cancellations.

Theorem

Integrate R Criterion with PS criterion:

Theorem. For $G_{\text{prev}} \subset G$, suppose

- G_{prev} is a Gröbner basis of $(f_1, f_2, \ldots, f_{m-1})$, and
- for all $1 \leq i < j \leq m$ such that $S(g_i, g_j) = u\,g_i - v\,g$ and $g_i \in (f_1, \ldots, f_m) \setminus (f_1, \ldots, f_{m-1})$
  we have (A) or (B) or (C) where
  (A) $u\,g_i$ or $v\,g_j$ has index m and satisfies PS wrt G_{prev};
  (B) $u\,g_i$ or $v\,g_j$ satisfies R;
  (C) $S(g_i, g_j) \xrightarrow[G]{*} 0$.

Then G is a Gröbner basis of $(f_1, f_2, \ldots, f_m)$.

Why? sketch

$$S := S(g_i, g_j) = \sum_{p \in F} h_p p$$

(A) $\Longrightarrow \exists$syzygy σ s.t. $S = S - \sigma$ has lower signature

(B) $\Longrightarrow \exists$syzygy σ s.t. $S = S - \sigma$ *and*

rewritable S-polys have lower signature

- new rep $\rightsquigarrow$ new cancellations? $\rightsquigarrow$ new S-polys
- new S-polys satisfy (A), (B), or (C), repeat
  - (A), (B), (C) *never* increase signature
  - (A), (B) decrease signature of rewritable
  - (C) lowers lcms
- finitely many S-polys $\Longrightarrow$ finitely many signatures
  $\rightsquigarrow$ "decrease" signature until no rewritable S-polys
  $\rightsquigarrow$ all must satisfy (C) $\Longrightarrow$ GB

Why? sketch

(Eder)

$$S(g_i, g_i) = h_1^{(1)} f_1 + \cdots + h_m^{(1)} f_m$$

$\downarrow$ (A), (B), (C)

$$S(g_i, g_j) = \mu_1 S(g_{i_1}, g_{j_1}) + h_1^{(2)} g_1 + \cdots + h_m^{(2)} g_m \qquad \exists \mu_1$$

$\downarrow$

$\vdots$

$$S(g_i, g_j) = \sum \mu_\lambda S(g_{i_\lambda}, g_{j_\lambda})$$

Why? sketch

(Perry)

$$S\left(g_i, g_i\right) = h_1^{(1)} f_1 + \cdots + h_m^{(1)} f_m$$

$$\downarrow \text{(A), (B), (C)}$$

$$S\left(g_i, g_i\right) = h_1^{(2)} g_1 + \cdots + h_m^{(2)} g_m$$

$$\downarrow$$

$$\vdots$$

$$S\left(f_i, f_j\right) = h_1^{(r)} g_1 + \cdots + h_m^{(r)} g_m$$

$$\mathrm{lt}\left(h_i^{(r)} g_i\right) \preceq \mathrm{lt}\left(S\left(f_i, f_j\right)\right) \quad \forall i = 1, 2, \ldots, m$$

Part 2

Implementing F5

Challenge

- understanding?
- pseudocode difficult to follow
  - unfamiliar notation
  - fatal & misleading errors
  - complex: subalgorithms, different cases
- some systems work with a broken implementation
  - false sense of confidence...
  - “stages interdependent” (Stegers)

Maple implementation

2007 Modification of "traditional" Buchberger algorithm

Maple programming language (Faugère: C)

excruciatingly slow

Faugère: "very easy"

- Stegers (MAGMA): "considerable effort"
- Perry: months

pseudocode & commentary:

http://www.math.usm.edu/perry/Research/F5Pseudocode.pdf

Maple implementation

2008 Slight modification to use matrix

not really F4-ish

slower!!!

gave up

Singular implementation

2008	basic modification of "traditional" Buchberger algorithm
	Singular programming language
	much faster than Maple
	Cyclic-6: <3 minutes vs. >15 minutes
	two days' work (correct pseudocode helps)
to-do	move to kernel
	sparse linear algebra?
	compiled C++ vs. interpreted Singular

Buchberger's Criteria?

Sometimes BC $\Longrightarrow$ (PS or R)

but not always!

Question: Can we improve F5 using BC?

Buchberger's Criteria?

Sometimes BC $\Longrightarrow$ (PS or R)

but not always!

Question: Can we improve F5 using BC?

NO

Buchberger's Criteria

Example: $(\mathbb{Z}_{29})$

$$f_1 = 3x^4y + 18xy^4 + 4x^3yz + 20xy^3z + 3x^2z^3$$
$$f_2 = 3x^3y^2 + 7x^2y^3 + 24y^2z^3$$
$$f_3 = 12xy^4 + 17x^4z + 27y^4z + 11x^3z^2$$

$S(f_1, f_4)$ "caught" by Buchberger's 2nd Criterion (via f_6)

BUT!

skipping $S(f_1, f_4)$ does *not* give a Gröbner basis

- 28 polynomials in result (vs. 30 if no skip)
- 66 S-polynomials do not reduce to zero
  - $S(f_1, f_4)$ is *not* one of these

Buchberger's Criteria

Why fail?

- certainly $S(f_1, f_4)$ *should* reduce to zero
  - $S(f_1, f_6)$, $S(f_4, f_6)$ reduce to zero
  - distinct lcms $\Longrightarrow$ no "three-way trap"
- triangularization: missed step!
  - Buchberger's Criteria ignore signature
  - signatures $x^2 z f_1$ and $y^3 f_2$
  - neither signature is considered if we skip
  - top-reduction: some reductions forbidden
  - unmodified F5: top-reduction of $S(f_1, f_4)$ stops with polynomial whose signature is $x^2 z f_1$

Note: in $\mathbb{Z}_{11}$, does not appear to terminate!

Other optimizations

Stegers, pg. 41

- compute *reduced* GB R_{prev} after each increment
- reduce, top-reduce wrt R_{prev}, not wrt G_{prev}
  - safe: smaller signatures in R_{prev}
- do not *replace* G_{prev} w/R_{prev}
  - "signature corruption"

Other optimizations

Stegers, pg. 41

- compute *reduced* GB R_{prev} after each increment
- reduce, top-reduce wrt R_{prev}, not wrt G_{prev}
  - safe: smaller signatures in R_{prev}
- do not *replace* G_{prev} w/R_{prev}
  - "signature corruption"

Yes, *but*...

Other optimizations

Can replace G_{prev} with $R_{\text{prev}} = (g_1, g_3, \ldots, g_M)$

- create appropriate signatures
  - compute basis for $\left(g_1, g_2, g_3, \ldots, g_M, f_m\right)$
  - signatures $1g_1$, $1g_2$, ..., $1g_{M+1}$ $\quad (g_{M+1} = f_m)$
  - also basis for $\left(f_1, f_2, \ldots, f_m\right)$!
  - no more signature corruption!
- create appropriate rules
  - $S\left(g_i, g_j\right) \xrightarrow[R_{\text{prev}}]{*} 0$
  - signatures for zero polynomial

Other optimizations

Can replace G_{prev} with $R_{\text{prev}} = (g_1, g_3, \ldots, g_M)$

- create appropriate signatures
  - compute basis for $(g_1, g_2, g_3, \ldots, g_M, f_m)$
  - signatures $\mathbf{1}g_1, \mathbf{1}g_2, \ldots, \mathbf{1}g_{M+1}$ $\quad (g_{M+1} = f_m)$
  - also basis for $(f_1, f_2, \ldots, f_m)$!
  - no more signature corruption!
- create appropriate rules
  - $S(g_i, g_j) \xrightarrow[R_{\text{prev}}]{*} 0$
  - signatures for zero polynomial

much more efficient than *published* F5

Other optimizations

Can replace G_{prev} with $R_{\text{prev}} = (g_1, g_3, \ldots, g_M)$

- create appropriate signatures
  - compute basis for $(g_1, g_2, g_3, \ldots, g_M, f_m)$
  - signatures $1g_1$, $1g_2$, …, $1g_{M+1}$ $\qquad (g_{M+1} = f_m)$
  - also basis for $(f_1, f_2, \ldots, f_m)$!
  - no more signature corruption!
- create appropriate rules
  - $S(g_i, g_j) \xrightarrow[R_{\text{prev}}]{*} 0$
  - signatures for zero polynomial

"A new, *more* efficient algorithm for computing Gröbner bases *with* reduction to zero"?

Some example systems

system	basic	reduce w/R_{prev}	compute w/R_{prev}
f633	161.85	160.86	132.24
Katsura6	13.242	12.78	8.74
Katsura7	111.03	97.1	54.3
Cyclic-6	161.35	159.5	118.74

(time in seconds, using Singular `timer`)

Interpreted Gebauer-Möller algorithm typically slower.

Other optimizations

- $\mathbb{Z}_2$: $f^2 = f$ (Bardet, Faugère, Salvy 2003)
- general: syzygies that lower signature?

Conclusion

- algorithm better understood (we hope)
  - termination? unresolved
- new, open-source implementations in interpreted languages
  - slow, okay, yes
  - compilation, linear algebra? $\rightsquigarrow$ significant speedup
- optimization: possibilities

Thanks

- Christian Eder (joint work)
- Jean-Charles Faugère
- Александр Семёнов
- Till Stegers
- Алексей Зобнин
- Technische Universität Kaiserslautern