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: algorithm to compute Gröbner bases of polynomial ideals

(J-C Faugère, 2002)
Remarks on Faugère’s F5 algorithm

John Perry

F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

Implementation
Why?
Where?
Two variants

Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination

Outline

1 F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

2 Implementation
Why?
Where?
Two variants

3 Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination
Gröbner bases?

Gröbner basis: “nice form” for generators of polynomial ideal

- “nice”: difficult questions

(B Buchberger, 1965)

Generalizes linear algebra

- Vector space: Gaussian elimination $\rightarrow$ echelon form

\[
\begin{array}{c}
\ast \ast \ast \ast = \ast \\
\ast \ast \ast \ast = \ast \\
\ast \ast \ast = \ast \\
\ast \ast = \ast \\
\ast = \ast \\
\end{array}
\rightarrow
\begin{array}{c}
\ast \ast \ast \ast = \ast \\
\ast \ast \ast = \ast \\
\ast \ast = \ast \\
\ast = \ast \\
\ast = \ast \\
\end{array}
\]

- Polynomial ring: Buchberger’s algorithm $\rightarrow$ Gröbner basis
Gröbner bases?

Gröbner basis: “nice form” for generators of polynomial ideal

- “nice”: difficult questions

(B Buchberger, 1965)

Generalizes linear algebra

- Vector space: Gaussian elimination $\rightarrow$ echelon form

\[
\begin{align*}
\{ & \begin{array}{cccc}
* & * & * & * \equiv * \\
* & * & * & * \equiv * \\
* & * & * & * \equiv * \\
* & * & * & * \equiv *
\end{array} \rightarrow \\
& \begin{array}{cccc}
* & * & * & * \equiv * \\
* & * & * & * \equiv * \\
* & * & * \equiv * \\
* & * \equiv *
\end{array}
\end{align*}
\]

- Polynomial ring: Buchberger’s algorithm $\rightarrow$ 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 lcm (lt$p$, 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=???)
Remarks on Faugère’s F5 algorithm

John Perry

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 lcm (lt$p, lt$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 $F \in \mathbb{F}[x_1, \ldots, x_n]^m$:

1. $G := F$

2. Consider all $p, q \in G$
   
   1. Compute $S := up - vq$
      
      ($u, p$ cancel lcm (lt$p, ltq$))
   
   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 lcm (lt $p$, 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=???)
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. $G = (xy + 1, y^2 + 1)$
   
   $S = y(xy + 1) - x(y^2 + 1) = y - x$
   
   No top-reduction

2. $G = (xy + 1, y^2 + 1, x - y)$
   
   $S = (xy + 1) - y(x - y) = 1 + y^2$
   
   Top-reduces to zero

   $S = x(y^2 + 1) - y^2(x - y) = x + y^3$
   
   Top-reduces to zero
Quick example

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

1. \( G = \langle xy + 1, y^2 + 1 \rangle \)
   
   1. \( S = y(xy + 1) - x(y^2 + 1) = y - x \)
      No top-reduction

2. \( G = \langle xy + 1, y^2 + 1, x - y \rangle \)
   
   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 \([xy + 1, y^2 + 1]\).

1. \[G = (xy + 1, y^2 + 1)\]
2. \[S = y(xy + 1) - x(y^2 + 1) = y - x\]
   No top-reduction

1. \[G = (xy + 1, y^2 + 1, x - y)\]
2. \[S = (xy + 1) - y(x - y) = 1 + y^2\]
   Top-reduces to zero
3. \[S = x(y^2 + 1) - y^2(x - y) = x + y^3\]
   Top-reduces to zero

\[\therefore \text{GB } (\langle xy + 1, y^2 + 1 \rangle) = (xy + 1, y^2 + 1, x - y).\]
Remarks on Faugère’s F5 algorithm

John Perry

Bottleneck

- Bottleneck
  - New polynomials → new information
  - Top-reduction to zero ↯ no new polynomial

  ↯ new information

- (100 − ε)% of time: verifying GB, not computing
- Top-reduction very, very expensive
Past work

- *Predict zero reductions*


- *Selection strategy*: Pick pairs in clever ways


- *Forbid some top-reductions*: Involution bases

  (V Gerdt-Y Blinkov 1998)

- *Homogenization*: $d$-Gröbner bases
Remarks on Faugère’s F5 algorithm

John Perry

F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

Implementation
Why?
Where?
Two variants

Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination

Outline

1 F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

2 Implementation
Why?
Where?
Two variants

3 Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination
Remarks on Faugère’s F5 algorithm

John Perry

F5: overview

F5: combined approach

- Homogenize
- \( d \)-Gröbner bases
- New point of view:
  - New way to predict zero reductions
  - New selection strategy

- Some systems: \textit{no zero reductions}!

“A new efficient algorithm for computing Gröbner bases without reduction to zero (\(F_5\))”
View from linear algebra

- Compute GB ⇔ 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:** \( G = \langle xy + h^2, y^2 + h^2 \rangle \)

\( d = 3 \):

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

Rows 2, 3 cancel…
Quick example, revisited

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

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

$d = 3$:

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

New! $g_3 = xb^2 - yb^2$
Quick example, revisited

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

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

\( d = 3 \):

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

linear dependence: \( xg_2 = g_3 + yg_1 \)
Quick example, revisited

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

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

\( d = 4: \)

\[
\begin{pmatrix}
    x^3 & x^2y & xy^2 & y^3 & y^4 & x^2h & xyh & y^2h & h^4 \\
    1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
    1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
    1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
    1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
    1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
    1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
    1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
    1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
    1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{pmatrix}
\]

linear dependence: \( x^2g_1, xyg_1, y^2g_1, h^2g_1 \)
Quick example, revisited

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

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

\( d = 4: \)

\[
\begin{pmatrix}
  x^3y & x^2y^2 & xy^3 & y^4 & x^2h^2 & xyh^2 & y^2h^2 & h^4 \\
  1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
  \end{pmatrix}
\]

\( x^2g_1 \quad xyg_1 \quad y^2g_1 \quad h^2g_1 \quad y^2g_2 \quad xg_3 \quad yg_3 \)

Rows 4, 7 cancel…
Remarks on Faugère’s F5 algorithm

John Perry

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:

\[
\begin{pmatrix}
 x^3y & x^2y^2 & xy^3 & y^4 & x^2h^2 & xyh^2 & y^2h^2 & h^4 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{pmatrix}
\]

Rows 4, 7 cancel… but we will not consider them!

**Why not?**

Later.
Remarks on Faugère’s F5 algorithm

John Perry

F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

Implementation
Why?
Where?
Two variants

Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination

Outline

1 F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

2 Implementation
Why?
Where?
Two variants

3 Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination
Remarks on Faugère’s F5 algorithm

John Perry

F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

Implementation
Why?
Where?
Two variants

Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination

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
- \( yg_3 \): \( g_3 = xg_2 - yg_1 \)
Remarks on Faugère’s F5 algorithm

John Perry

F5

Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

Implementation

Why?
Where?
Two variants

Termination (?)

The difficulty
Faugère’s original argument
Non-terminating example...terminates!
Variants that guarantee termination

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
• \(yg_3\): \(g_3 = xg_2 - yg_1\)

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

\(\therefore\) \(\text{Sig}(yg_3) = xyg_2\).
Signatures: Observations

- \( \text{Sig}(p) = tg_i ? \)
  - \( 1 \leq i \leq m \)
  - \( g = b_1 g_1 + \cdots + b_{i-1} g_{i-1} + (t + \cdots) g_i \) (inputs: \((g_1, \ldots, g_m)\)) (\( \exists h_1, \ldots, b_i, \text{lt}(h_i) = t \))

- this definition \( \neq \) Faugère’s definition

- “easy” record-keeping: list of rules
- “easily” reject certain useless pairs:
  - Use \( yg_3 \ w/\text{sig}\ xyg_2, \not xyg_2 \)
  - Use \( xg_3 \ w/\text{sig}\ x^2g_2, \not x^2g_2 \)
  - ...

- Criterion “Rewritten”

Signatures: Observations

- \( \text{Sig}(p) = tg_i ? \)
  - \( 1 \leq i \leq m \)
  - \( g = b_1g_1 + \cdots + b_{i-1}g_{i-1} + (t + \cdots)g_i \) (inputs: \( (g_1, \ldots, g_m) \)) \( (\exists h_1, \ldots, h_i, \text{lt}(h_i) = t) \)
  - 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 \( x^2g_2 \), not \( x^2g_2 \)
  - …

- Criterion “Rewritten”

Faugère’s characterization

Theorem (Faugère, 2002)

\[(A) \iff (B) \text{ where} \]

\[(A) \text{ } G \text{ a Gröbner basis} \]
\[(B) \forall p, q \in G \text{ where} \]
\[\begin{itemize}
    \item \text{uSig}(p), \text{vSig}(q) \text{ not rewritable, and} \\
    \item \text{uSig}(p), \text{vSig}(q) \text{ minimal}
\end{itemize} \]

\text{S-polynomial } up - vq \text{ top-reduces to zero w/out changing signature}

(highly paraphrased, slightly generalized)
Remarks on Faugère’s F5 algorithm

John Perry

Outline

1. F5
   - Gröbner bases: review
   - Rough idea
   - Signatures
   - Predicting zero reductions
   - The algorithm

2. Implementation
   - Why?
   - Where?
   - Two variants

3. Termination (?)
   - The difficulty
   - Faugère’s original argument
   - Non-terminating example . . . terminates!
   - Variants that guarantee termination
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 to predict zero reductions?

- Recall

\[
\begin{pmatrix}
x^3 y & x^2 y^2 & xy^3 & y^4 & x^2 h^2 & xyh^2 & y^2 h^2 & h^4 \\
\end{pmatrix}
\]

We did not cancel. *Why not?*

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

How?
Remarks on Faugère’s F5 algorithm

John Perry

F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

Implementation
Why?
Where?
Two variants

Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination

Faugère’s criterion

Theorem
If
• \( u \text{Sig}(p) = u g_i \); and
• \( \text{lt}(q) \mid u, \exists q \in \text{GB}_{\text{prev}} (g_1, \ldots, g_{i-1}) \);
then \( u \text{Sig}(p) \) is not minimal.

Definition
\( \text{FC}(u \text{Sig}(p)) : \text{lt}(q) \mid u, \exists q \in \text{G}_{\text{prev}} \)

Corollary
In S-polynomial \( u p - vq \),
if \( \text{FC}(u \text{Sig}(p)) \) or \( \text{FC}(v \text{Sig}(q)) \)
then we need not compute S.
Remarks on Faugère’s F5 algorithm

John Perry

F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

Implementation
Why?
Where?
Two variants

Termination (??)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination

Faugère’s criterion

Theorem

If

- \( u \text{Sig}(p) = u g_i \); and
- \( \text{lt}(q) | u, \exists q \in \text{GB}_{\text{prev}}(g_1, \ldots, g_{i-1}) \);

then \( u \text{Sig}(p) \) is not minimal.

Definition

\( \text{FC}(u \text{Sig}(p)) : \text{lt}(q) | u, \exists q \in \text{GB}_{\text{prev}} \)

Corollary

In S-polynomial \( u p - v q \),

if \( \text{FC}(u \text{Sig}(p)) \) or \( \text{FC}(v \text{Sig}(q)) \)

then we need not compute S.
Remarks on Faugère’s F5 algorithm

John Perry

F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

Implementation
Why?
Where?
Two variants

Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination

In the example...

- Recall

\[
\begin{pmatrix}
  x^3 y & x^2 y^2 & xy^3 & y^4 & x^2 h^2 & xy h^2 & y^2 h^2 & h^4 \\
  \vdots & & & & & & & \\
  1 & -1 & h^2 g_1 \\
  \vdots & & & & & & & \\
  1 & -1 & y g_3
\end{pmatrix}
\]

- \( G_{\text{prev}} = (g_1) \)
- \( \text{Sig}(g_3) = x g_2 \)

- \( y \text{Sig}(g_3) = xy g_2, \text{ and } \text{lt}(g_1) | xy... \)

\[ \text{FC} \implies \text{no need to compute S-polynomial} \]

Why?
In the example...

- Recall

\[
\begin{pmatrix}
x^3 y & x^2 y^2 & xy^3 & y^4 & x^2 h^2 & xy h^2 & y^2 h^2 & h^4 \\
1 & -1 & h^2 g_1 \\
1 & -1 & yg_3
\end{pmatrix}
\]

- \( G_{\text{prev}} = (g_1) \)
- \( \text{Sig}(g_3) = xg_2 \)

\( \gamma \text{Sig}(g_3) = xyg_2 \), and \( \text{lt}(g_1) | xy \ldots \)

FC \( \Rightarrow \) no need to compute \( S \)-polynomial

Why?
Why? Trivial syzygies

Recall $y g_3 = y [x g_2 - y g_1]$...
Why? Trivial syzygies

Recall $y g_3 = y [x g_2 - y g_1] \ldots$

$\therefore y g_3 = y [x g_2 - y g_1]$

$= x y g_2 - y^2 g_1$
Why? Trivial syzygies

Recall \( yg_3 = y[xg_2 - yg_1] \) …

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

Trivially \( g_1g_2 - g_2g_1 = 0 \).
Why? Trivial syzygies

Recall \( yg_3 = y[xg_2 - yg_1] \)...

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

Trivially \( g_1g_2 - g_2g_1 = 0 \).

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

Recall \( yg_3 = y[g_2 - g_1] \)…

\[
\therefore yg_3 = y[g_2 - g_1] \\
= xyg_2 - y^2 g_1
\]

Trivially \( g_1g_2 - g_2g_1 = 0 \).

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

\( \text{Sig}(yg_3) \) not minimal!
Remarks on Faugère’s F5 algorithm

John Perry

Outline

1 F5
   Gröbner bases: review
   Rough idea
   Signatures
   Predicting zero reductions
   The algorithm

2 Implementation
   Why?
   Where?
   Two variants

3 Termination (?)
   The difficulty
   Faugère’s original argument
   Non-terminating example...terminates!
   Variants that guarantee termination
The F5 Algorithm

1. Each stage: Incremental strategy
   1. Compute $\text{GB}(g_1)$
   2. Compute $\text{GB}(g_1, g_2)$
   3. ...

2. $d$-GB’s $\rightsquigarrow \text{GB}(g_1, \ldots, g_i)$

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
Remarks on Faugère’s F5 algorithm

John Perry

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 $\leadsto$ 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
Remarks on Faugère’s F5 algorithm

John Perry

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.
Remarks on Faugère’s F5 algorithm

John Perry

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)
Remarks on Faugère’s F5 algorithm

John Perry

1. F5
   Gröbner bases: review
   Rough idea
   Signatures
   Predicting zero reductions
   The algorithm

2. Implementation
   Why?
   Where?
   Two variants

3. Termination (?)
   The difficulty
   Faugère’s original argument
   Non-terminating example... terminates!
   Variants that guarantee termination
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?
Remarks on Faugère’s F5 algorithm

John Perry

1. F5
   - Gröbner bases: review
   - Rough idea
   - Signatures
   - Predicting zero reductions
   - The algorithm

2. Implementation
   - Why?
   - Where?
   - Two variants

3. Termination (?)
   - The difficulty
   - Faugère’s original argument
   - Non-terminating example... terminates!
   - Variants that guarantee termination
Implementations (1)

- **Faugère (2002)**
  - C, interfaces w/Maple
  - *Very* fast
  - Several variants: F5, F5’, F5”, …?
  - Source code not publicly available, binary download

- **Stegers (2005)**
  - Interpreted Magma code
  - Respectable timings
  - Variant “F5R”

- **Others**
  - Unstable implementations
  - Magma implementation?
Remarks on Faugère’s F5 algorithm

John Perry

Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

Implementation
Why?
Where?
Two variants

Termination (??)
The difficulty
Faugère’s original argument
Non-terminating example...terminates!
Variants that guarantee termination

Implementations (2)

- Perry (2007)
  - Interpreted Maple code
  - Embarassingly slow
  - Source code publicly available

  - Interpreted Singular code
  - Respectable timings
  - New variant “F5C”
  - http://www.math.usm.edu/perry/research.html
Remarks on Faugère’s F5 algorithm

John Perry

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)
Remarks on Faugère’s F5 algorithm

John Perry

F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

Implementation
Why?
Where?
Two variants

Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example…terminates!
Variants that guarantee termination

Outline

1  F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

2  Implementation
Why?
Where?
Two variants

3  Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example…terminates!
Variants that guarantee termination
Remarks on Faugère’s F5 algorithm

John Perry

F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

Implementation
Why?
Where?
Two variants

Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination

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
Remarks on Faugère’s F5 algorithm

John Perry

F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

Implementation
Why?
Where?
Two variants

Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example… terminates!
Variants that guarantee termination

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 $\langle f_1, \ldots, f_i \rangle$
- Compute RGB $B$ of $\langle G \rangle$
- 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

(usual F5)  
(easy: interreduce $G$)
F5C (Eder and Perry, 2008–2009)

- Compute GB $G$ of $\langle f_1, \ldots, f_i \rangle$
- Compute RGB $B$ of $\langle G \rangle$
- 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
Remarks on Faugère’s F5 algorithm

John Perry

#Critical pairs, #Polynomials in variants

<table>
<thead>
<tr>
<th>$i$</th>
<th>$#G_{\text{curr}}$</th>
<th>$\max {#P_d}$</th>
<th>$i$</th>
<th>$#G_{\text{curr}}$</th>
<th>$\max {#P_d}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>N/A</td>
<td>2</td>
<td>2</td>
<td>N/A</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>16</td>
<td>4</td>
<td>5</td>
<td>15</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>32</td>
<td>8</td>
<td>6</td>
<td>29</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>60</td>
<td>17</td>
<td>7</td>
<td>51</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>132</td>
<td>29</td>
<td>8</td>
<td>109</td>
<td>29</td>
</tr>
<tr>
<td>9</td>
<td>524</td>
<td>89</td>
<td>9</td>
<td>472</td>
<td>71</td>
</tr>
<tr>
<td>10</td>
<td>1165</td>
<td>276</td>
<td>10</td>
<td>778</td>
<td>89</td>
</tr>
</tbody>
</table>
Remarks on Faugère’s F5 algorithm

John Perry

F5
Grobner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

Implementation
Why?
Where?
Two variants

Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination

#Reductions

<table>
<thead>
<tr>
<th>variant:</th>
<th>F5</th>
<th>F5R</th>
<th>F5C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Katsura-5</td>
<td>346</td>
<td>289</td>
<td>222</td>
</tr>
<tr>
<td>Katsura-6</td>
<td>8,357</td>
<td>2,107</td>
<td>1,383</td>
</tr>
<tr>
<td>Katsura-7</td>
<td>1,025,408</td>
<td>24,719</td>
<td>10,000</td>
</tr>
<tr>
<td>Cyclic-5</td>
<td>441</td>
<td>457</td>
<td>415</td>
</tr>
<tr>
<td>Cyclic-6</td>
<td>36,139</td>
<td>17,512</td>
<td>10,970</td>
</tr>
</tbody>
</table>

(Top-reduction, normal forms)

(Many more in Gebauer-Möller: > 1,500,000 in Cyclic-6)
Remarks on Faugère’s F5 algorithm

John Perry

1 F5
   Gröbner bases: review
   Rough idea
   Signatures
   Predicting zero reductions
   The algorithm

2 Implementation
   Why?
   Where?
   Two variants

3 Termination (?)
   The difficulty
   Faugère’s original argument
   Non-terminating example... terminates!
   Variants that guarantee termination
Remarks on Faugère’s F5 algorithm

John Perry

F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

Implementation
Why?
Where?
Two variants

Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination

Termination: the difficulty

Termination?

• Buchberger: ACC $\implies$ S-polys reduce to zero eventually

• Faugère: S-polys w/ minimal signatures computed, but...
Remarks on Faugère’s F5 algorithm

Termination: the difficulty

Termination?

- Buchberger: ACC $\implies$ 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?
Remarks on Faugère’s F5 algorithm

John Perry

Outline

1. F5
   - Gröbner bases: review
   - Rough idea
   - Signatures
   - Predicting zero reductions
   - The algorithm

2. Implementation
   - Why?
   - Where?
   - Two variants

3. Termination (?)
   - The difficulty
     - Faugère’s original argument
     - Non-terminating example...terminates!
     - Variants that guarantee termination
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
- \(\leadsto\) 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 $p_2$
Remarks on Faugère’s F5 algorithm

John Perry

F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

Implementation
Why?
Where?
Two variants

Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination

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 \( \rightarrow \) new rewrite rule
- \( p_3 \) top-reducible by \( p_1 \)? \textit{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 $\implies$ 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 \(\Rightarrow\) 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...
Remarks on Faugère’s F5 algorithm

John Perry

F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

Implementation
Why?
Where?
Two variants

Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination

Outline

1 F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

2 Implementation
Why?
Where?
Two variants

3 Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination
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 $\Rightarrow$ termination!

(R Dellaca-J Gash-J Perry, 2009)
Remarks on Faugère’s F5 algorithm

John Perry

F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

Implementation
Why?
Where?
Two variants

Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination

Outline

1 F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

2 Implementation
Why?
Where?
Two variants

3 Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination
Remarks on Faugère’s F5 algorithm

John Perry

F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

Implementation
Why?
Where?
Two variants

Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination

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?
Gash (2008 PhD Dissertation)

- Redundant polynomials $\mapsto$ 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.
Remarks on Faugère’s F5 algorithm

John Perry

F5
Gröbner bases: review
Rough idea
Signatures
Predicting zero reductions
The algorithm

Implementation
Why?
Where?
Two variants

Termination (?)
The difficulty
Faugère’s original argument
Non-terminating example... terminates!
Variants that guarantee termination

Thank you!