Signature-based Gröbner Basis Algorithms

Christian Eder

joint work with

Jean-Charles Faugère, Bjarke Hammersholt Roune, John Perry
and Justin Gash

GBRELA2013 - Hagenberg, Austria

September 03 – 06, 2013

How to detect zero reductions in advance?

Example

Let $I = \langle g_1, g_2 \rangle \in \mathbb{Q}[x, y, z]$, $\mathbf{g_1 = xy - z^2}$, $\mathbf{g_2 = y^2 - z^2}$.
$<$ denotes the reverse lexicographical ordering.

How to detect zero reductions in advance?

Example

Let $I = \langle g_1, g_2 \rangle \in \mathbb{Q}[x, y, z]$, $\mathbf{g_1 = xy - z^2}$, $\mathbf{g_2 = y^2 - z^2}$.
$<$ denotes the reverse lexicographical ordering.

$$\begin{aligned} \text{spol}(g_2, g_1) &= xg_2 - yg_1 = \mathbf{xy^2} - xz^2 - \mathbf{xy^2} + yz^2 \\ &= -xz^2 + yz^2. \end{aligned}$$

Thus it reduces to $\mathbf{g_3 = xz^2 - yz^2}$ w.r.t. G.

How to detect zero reductions in advance?

Example

Let $I = \langle g_1, g_2 \rangle \in \mathbb{Q}[x, y, z]$, $\mathbf{g_1} = \mathbf{xy} - \mathbf{z^2}$, $\mathbf{g_2} = \mathbf{y^2} - \mathbf{z^2}$.
$<$ denotes the reverse lexicographical ordering.

$$\begin{aligned}\text{spol}(g_2, g_1) &= xg_2 - yg_1 = \mathbf{xy^2} - xz^2 - \mathbf{xy^2} + yz^2 \\ &= -xz^2 + yz^2.\end{aligned}$$

Thus it reduces to $\mathbf{g_3} = \mathbf{xz^2} - \mathbf{yz^2}$ w.r.t. G.

$$\text{spol}(g_3, g_1) = \mathbf{xyz^2} - y^2z^2 - \mathbf{xyz^2} + z^4 = -y^2z^2 + z^4.$$

How to detect zero reductions in advance?

Example

Let $I = \langle g_1, g_2 \rangle \in \mathbb{Q}[x, y, z]$, $\mathbf{g_1 = xy - z^2}$, $\mathbf{g_2 = y^2 - z^2}$.
$<$ denotes the reverse lexicographical ordering.

$$\begin{aligned} \mathrm{spol}(g_2, g_1) &= xg_2 - yg_1 = \mathbf{xy^2} - xz^2 - \mathbf{xy^2} + yz^2 \\ &= -xz^2 + yz^2. \end{aligned}$$

Thus it reduces to $\mathbf{g_3 = xz^2 - yz^2}$ w.r.t. G.

$$\mathrm{spol}(g_3, g_1) = \mathbf{xyz^2} - y^2z^2 - \mathbf{xyz^2} + z^4 = -y^2z^2 + z^4.$$

We can reduce further using $z^2 g_2$:

$$-y^2z^2 + z^4 + y^2z^2 - z^4 = 0.$$

How to detect zero reductions in advance?

Example

Let $I = \langle g_1, g_2 \rangle \in \mathbb{Q}[x, y, z]$, $\mathbf{g_1 = xy - z^2}$, $\mathbf{g_2 = y^2 - z^2}$.
$<$ denotes the reverse lexicographical ordering.

$$\begin{aligned}\text{spol}(g_2, g_1) &= xg_2 - yg_1 = \mathbf{xy^2} - xz^2 - \mathbf{xy^2} + yz^2 \\ &= -xz^2 + yz^2.\end{aligned}$$

Thus it reduces to $\mathbf{g_3 = xz^2 - yz^2}$ w.r.t. G.

$$\text{spol}(g_3, g_1) = \mathbf{xyz^2} - y^2z^2 - \mathbf{xyz^2} + z^4 = -y^2z^2 + z^4.$$

We can reduce further using $z^2 g_2$:

$$-y^2z^2 + z^4 + y^2z^2 - z^4 = 0.$$

How to get rid of this zero reduction?

Let $I = \langle f_1, \ldots, f_m \rangle$.
Idea: Give each $f \in I$ a bit more structure:

Let $I = \langle f_1, \ldots, f_m \rangle$.
Idea: Give each $f \in I$ a bit more structure:

1. Let R^m be generated by $e_1, \ldots, e_m$ and let $\prec$ be a compatible monomial order on the monomials of R^m.

Let $I = \langle f_1, \ldots, f_m \rangle$.
Idea: Give each $f \in I$ a bit more structure:

1. Let R^m be generated by $e_1, \ldots, e_m$ and let $\prec$ be a compatible monomial order on the monomials of R^m.

2. Let $\alpha \mapsto \overline{\alpha} : R^m \to R$ such that $\overline{e}_i = f_i$ for all i.

Let $I = \langle f_1, \ldots, f_m \rangle$.
Idea: Give each $f \in I$ a bit more structure:

1. Let R^m be generated by $e_1, \ldots, e_m$ and let $\prec$ be a compatible monomial order on the monomials of R^m.
2. Let $\alpha \mapsto \overline{\alpha} : R^m \to R$ such that $\overline{e}_i = f_i$ for all i.
3. Each $f \in I$ can be represented via some $\alpha \in R^m$: $f = \overline{\alpha}$

Let $I = \langle f_1, \ldots, f_m \rangle$.
Idea: Give each $f \in I$ a bit more structure:

1. Let R^m be generated by $e_1, \ldots, e_m$ and let $\prec$ be a compatible monomial order on the monomials of R^m.
2. Let $\alpha \mapsto \overline{\alpha} : R^m \to R$ such that $\overline{e}_i = f_i$ for all i.
3. Each $f \in I$ can be represented via some $\alpha \in R^m$: $f = \overline{\alpha}$
4. **A signature** of f is given by $\mathfrak{s}(f) = \mathrm{lt}_\prec(\alpha)$ where $f = \overline{\alpha}$.

Our example again – now with signatures and $\prec_{\text{pot}}$

$$
\begin{aligned}
g_1 &= xy - z^2,\ \mathfrak{s}(g_1) = e_1,\\
g_2 &= y^2 - z^2,\ \mathfrak{s}(g_2) = e_2,\\
g_3 &= \text{spol}(g_2, g_1) = xg_2 - yg_1\\
\Rightarrow \mathfrak{s}(g_3) &= x\,\mathfrak{s}(g_2) = xe_2.
\end{aligned}
$$

Our example again – now with signatures and $\prec_{\text{pot}}$

$$
\begin{aligned}
g_1 &= xy - z^2,\ \mathfrak{s}(g_1) = e_1,\\
g_2 &= y^2 - z^2,\ \mathfrak{s}(g_2) = e_2,\\
g_3 &= \operatorname{spol}(g_2, g_1) = xg_2 - yg_1\\
\Rightarrow \mathfrak{s}(g_3) &= x\,\mathfrak{s}(g_2) = xe_2.
\end{aligned}
$$

$\operatorname{spol}(g_3, g_1) = yg_3 - z^2 g_1$:

$$\mathfrak{s}\left(\operatorname{spol}(g_3, g_1)\right) = y\,\mathfrak{s}(g_3) = xye_2.$$

Our example again – now with signatures and $\prec_{\text{pot}}$

$$
\begin{aligned}
g_1 &= xy - z^2,\ \mathfrak{s}(g_1) = e_1,\\
g_2 &= y^2 - z^2,\ \mathfrak{s}(g_2) = e_2,\\
g_3 &= \operatorname{spol}(g_2, g_1) = xg_2 - yg_1\\
\Rightarrow \mathfrak{s}(g_3) &= x\,\mathfrak{s}(g_2) = xe_2.
\end{aligned}
$$

$\operatorname{spol}(g_3, g_1) = yg_3 - z^2 g_1$:

$$\mathfrak{s}\left(\operatorname{spol}(g_3, g_1)\right) = y\,\mathfrak{s}(g_3) = xye_2.$$

Note that $\mathfrak{s}\left(\operatorname{spol}(g_3, g_1)\right) = xy\,e_2$ and $\operatorname{lm}(g_1) = xy$.

Our example again – now with signatures and $\prec_{\text{pot}}$

$$
\begin{aligned}
g_1 &= xy - z^2,\ \mathfrak{s}(g_1) = e_1,\\
g_2 &= y^2 - z^2,\ \mathfrak{s}(g_2) = e_2,\\
g_3 &= \operatorname{spol}(g_2, g_1) = xg_2 - yg_1\\
\Rightarrow \mathfrak{s}(g_3) &= x\,\mathfrak{s}(g_2) = xe_2.
\end{aligned}
$$

$\operatorname{spol}(g_3, g_1) = yg_3 - z^2 g_1$:

$$\mathfrak{s}\left(\operatorname{spol}(g_3, g_1)\right) = y\,\mathfrak{s}(g_3) = xye_2.$$

Note that $\mathfrak{s}\left(\operatorname{spol}(g_3, g_1)\right) = xy\,e_2$ and $\operatorname{lm}(g_1) = xy$.

$\Rightarrow$ **We know that $\operatorname{spol}(g_3, g_1)$ reduces to zero w.r.t. G.**

How do we know this?

General idea: Per signature we only need to compute 1 element for G.

How do we know this?

General idea: Per signature we only need to compute 1 element for G.

> Several elements with the same signature?

How do we know this?

General idea: Per signature we only need to compute 1 element for G.

Several elements with the same signature?

↓

Choose 1 and remove the others.

How do we know this?

General idea: Per signature we only need to compute 1 element for G.

Several elements with the same signature?

↓

Choose 1 and remove the others.

Our goal: Make good choices.

How do we know this?

General idea: Per signature we only need to compute 1 element for G.

Several elements with the same signature?

↓

Choose 1 and remove the others.

Our goal: Make good choices.

Our task: Keep signatures correct.

$\alpha \in R^m$ stores all data needed:

- Polynomial $\overline{\alpha}$ with leading term $\mathrm{lt}\,(\overline{\alpha})$.
- Signature $\left[\,\mathfrak{s}(\overline{\alpha}) = \right]\mathfrak{s}(\alpha) = \mathrm{lt}\,(\alpha)$.

Think in the module

$\alpha \in R^m$ stores all data needed:

- Polynomial $\overline{\alpha}$ with leading term $\operatorname{lt}(\overline{\alpha})$.
- Signature $\left[\, \mathfrak{s}(\overline{\alpha}) = \right] \mathfrak{s}(\alpha) = \operatorname{lt}(\alpha)$.

Conventions:

- $\alpha \in R^m$ with $\overline{\alpha} = 0$ is a syzygy.
- $\mathfrak{s}$-reduction $\widehat{=}$ polynomial reduction **while retaining signature**
- $\mathfrak{s}$-reductions are always w.r.t. a finite basis $\mathcal{G} \subset R^m$.

- $\mathcal{G}$ is a **signature-based Gröbner Basis in signature** T if all $\alpha \in R^m$ with $\mathfrak{s}(\alpha) = T$ $\mathfrak{s}$-reduce to zero w.r.t. $\mathcal{G}$.
- $\mathcal{G}$ is a **signature-based Gröbner Basis** if $\mathcal{G}$ is a signature-based Gröbner Basis in all signatures
- If $\mathcal{G}$ is a signature-based Gröbner Basis then $\{\overline{\alpha} \mid \alpha \in \mathcal{G}\}$ is a Gröbner Basis for $\langle f_1, \ldots, f_m \rangle$.

- $\mathcal{G}$ is a **signature-based Gröbner Basis in signature** T if all $\alpha \in R^m$ with $\mathfrak{s}(\alpha) = T$ $\mathfrak{s}$-reduce to zero w.r.t. $\mathcal{G}$.
- $\mathcal{G}$ is a **signature-based Gröbner Basis** if $\mathcal{G}$ is a signature-based Gröbner Basis in all signatures
- If $\mathcal{G}$ is a signature-based Gröbner Basis then $\{\overline{\alpha} \mid \alpha \in \mathcal{G}\}$ is a Gröbner Basis for $\langle f_1, \ldots, f_m \rangle$.

Remark

In the following we need one detail from signature-based Gröbner Basis computations:

The pair set is ordered by increasing signature.

$$\mathfrak{s}(\alpha) = \mathfrak{s}(\beta) \implies \text{Compute 1, remove 1.}$$

Signature-based criteria

$$\mathfrak{s}(\alpha) = \mathfrak{s}(\beta) \implies \text{Compute 1, remove 1.}$$

Sketch of proof

1. $\mathfrak{s}(\alpha - \beta) \prec \mathfrak{s}(\alpha), \mathfrak{s}(\beta)$.
2. All S-pairs are handled by increasing signature.
   $\Rightarrow$ All relatons $\prec \mathfrak{s}(\alpha)$ are known:

$$\alpha = \beta + \text{ elements of smaller signature}$$

$\square$

S-pairs in signature T

S-pairs in signature T

What are all possible configurations to reach signature T?

S-pairs in signature T

$$\mathfrak{R}_T = \left\{ a\alpha \mid \alpha \text{ handled by the algorithm and } \mathfrak{s}(a\alpha) = T \right\}$$

What are all possible configurations to reach signature T?

Signature-based criteria

S-pairs in signature T

$$\mathfrak{R}_T = \left\{ a\alpha \mid \alpha \text{ handled by the algorithm and } \mathfrak{s}(a\alpha) = T \right\}$$

What are all possible configurations to reach signature T?

Define an order on $\mathfrak{R}_T$ and choose the maximal element.

Special cases

$$\mathfrak{R}_T = \left\{ a\alpha \mid \alpha \text{ handled by the algorithm and } \mathfrak{s}(a\alpha) = T \right\}$$

$$\mathfrak{R}_T = \left\{ a\alpha \mid \alpha \text{ handled by the algorithm and } \mathfrak{s}(a\alpha) = T \right\}$$

1. If $a\alpha$ is a syzygy $\Longrightarrow$ Go on to next signature.

Special cases

$$\mathfrak{R}_T = \Big\{ a\alpha \mid \alpha \text{ handled by the algorithm and } \mathfrak{s}(a\alpha) = T \Big\}$$

1. If $a\alpha$ is a syzygy $\implies$ Go on to next signature.
2. If $a\alpha$ is not part of an S-pair $\implies$ Go on to next signature.

$$\mathfrak{R}_T = \left\{ a\alpha \mid \alpha \text{ handled by the algorithm and } \mathfrak{s}(a\alpha) = T \right\}$$

1. If $a\alpha$ is a syzygy $\Longrightarrow$ Go on to next signature.
2. If $a\alpha$ is not part of an S-pair $\Longrightarrow$ Go on to next signature.

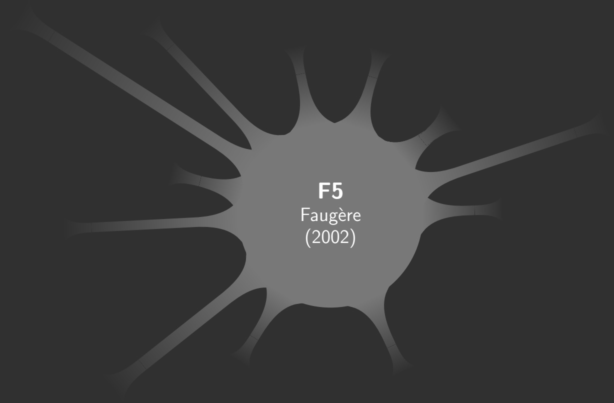
Revisiting our example with $\prec_{\text{pot}}$

$\mathfrak{s}\left(\text{spol}(g_3, g_1)\right) = xye_2$

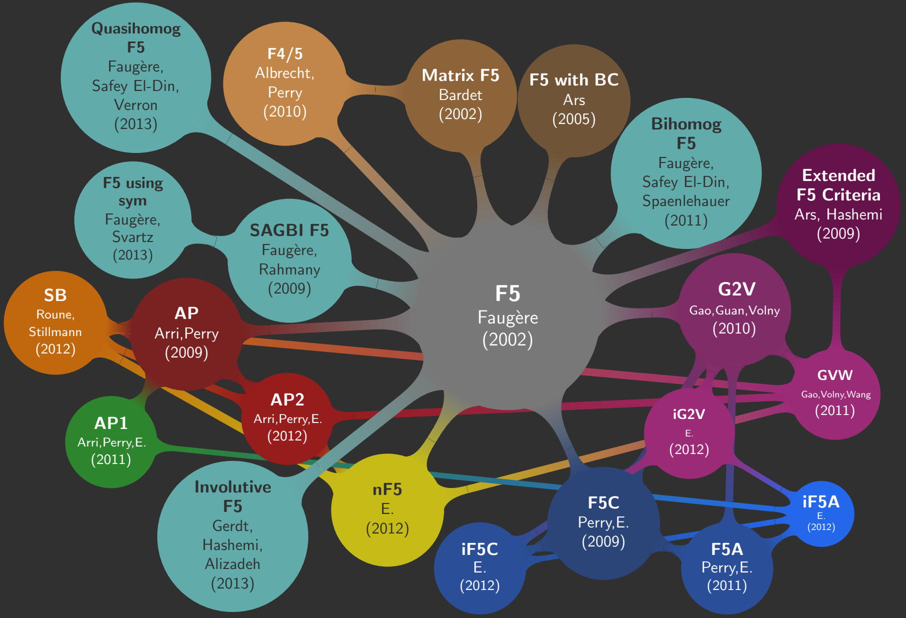
$$\left.\begin{array}{l} g_1 = xy - z^2 \\ g_2 = y^2 - z^2 \end{array}\right\} \Rightarrow \text{psyz}(g_2, g_1) = g_1 e_2 - g_2 e_1 = xye_2 + \dots$$

- The basic problem
- Signature basics
- Signature-based criteria
- **A decade in signature-based Gröbner Basis algorithms**

A decade in signature-based Gröbner Basis algorithms



A decade in signature-based Gröbner Basis algorithms



References

[AP10] M. Albrecht und J. Perry. F4/5

[AP11] A. Arri und J. Perry. The F5 Criterion revised

[EGP11] C. Eder, J. Gash and J. Perry. Modifying Faugère's F5 Algorithm to ensure termination

[EP10] C. Eder and J. Perry. F5C: A variant of Faugère's F5 Algorithm with reduced Gröbner bases

[EP11] C. Eder and J. Perry. Signature-based algorithms to compute Gröbner bases

[ER13] C. Eder and B. H. Roune. Signature Rewriting in Gröbner Basis Computation

[F99] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F4)

[F02] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero F_5

[FJ03] J.-C. Faugère and A. Joux. Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases

[FL10] J.-C. Faugère and S. Lachartre. Parallel Gaussian Elimination for Gröbner bases computations in finite fields

[GGV10] S. Gao, Y. Guan and F. Volny IV. A New Incremental Algorithm for Computing Gröbner Bases

[GVW11] S. Gao, F. Volny IV and M. Wang. A New Algorithm For Computing Grobner Bases

[RS12] B. H. Roune and M. Stillman. Practical Gröbner Basis Computation