Signature-based algorithms to compute Gröbner bases

Christian Eder
(joint work with John Perry)

University of Kaiserslautern

June 29, 2011

The following section is about

An example of zero reduction

Example

Given $g_1 = xy - z^2$, $g_2 = y^2 - z^2$,we can compute

$$\mathrm{Spol}(g_2, g_1) = \mathbf{xy^2} - xz^2 - \mathbf{xy^2} + yz^2 = -xz^2 + yz^2.$$

An example of zero reduction

Example

Given $g_1 = xy - z^2$, $g_2 = y^2 - z^2$,we can compute

$$\mathrm{Spol}(g_2, g_1) = \mathbf{xy^2} - xz^2 - \mathbf{xy^2} + yz^2 = -xz^2 + yz^2.$$

We get a new element $g_3 = xz^2 - yz^2$ for G.

An example of zero reduction

Example

Given $g_1 = xy - z^2$, $g_2 = y^2 - z^2$, we can compute

$$\mathrm{Spol}(g_2, g_1) = \mathbf{xy^2} - xz^2 - \mathbf{xy^2} + yz^2 = -xz^2 + yz^2.$$

We get a new element $g_3 = xz^2 - yz^2$ for G.

Let us compute $\mathrm{Spol}(g_3, g_1)$ next:

An example of zero reduction

Example

Given $g_1 = xy - z^2$, $g_2 = y^2 - z^2$, we can compute

$$\mathrm{Spol}(g_2, g_1) = \mathbf{xy^2} - xz^2 - \mathbf{xy^2} + yz^2 = -xz^2 + yz^2.$$

We get a new element $g_3 = xz^2 - yz^2$ for G.
Let us compute $\mathrm{Spol}(g_3, g_1)$ next:

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

An example of zero reduction

Example

Given $g_1 = xy - z^2$, $g_2 = y^2 - z^2$, we can compute

$$\mathrm{Spol}(g_2, g_1) = \mathbf{xy^2} - xz^2 - \mathbf{xy^2} + yz^2 = -xz^2 + yz^2.$$

We get a new element $g_3 = xz^2 - yz^2$ for G.
Let us compute $\mathrm{Spol}(g_3, g_1)$ next:

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

Now we can reduce further with $z^2 g_2$:

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

Example

Given $g_1 = xy - z^2$, $g_2 = y^2 - z^2$, we can compute

$$\mathrm{Spol}(g_2, g_1) = \mathbf{xy^2} - xz^2 - \mathbf{xy^2} + yz^2 = -xz^2 + yz^2.$$

We get a new element $g_3 = xz^2 - yz^2$ for G.

Let us compute $\mathrm{Spol}(g_3, g_1)$ next:

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

Now we can reduce further with $z^2 g_2$:

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

$\Rightarrow$ **How to detect zero reductions in advance?**

Let $I = \langle f_1, \ldots, f_m \rangle$. The idea is to give each polynomial during the computations of the algorithm a so-called **signature**:

Let $I = \langle f_1, \ldots, f_m \rangle$. The idea is to give each polynomial during the computations of the algorithm a so-called **signature**:

1. Let $e_1, \ldots, e_m \in R^m$ be canonical generators such that $\pi : R^m \to R$: $\pi(e_i) = f_i$ for all i.

Signatures of polynomials

Let $I = \langle f_1, \ldots, f_m \rangle$. The idea is to give each polynomial during the computations of the algorithm a so-called **signature**:

1. Let $e_1, \ldots, e_m \in R^m$ be canonical generators such that $\pi : R^m \to R$: $\pi(e_i) = f_i$ for all i.
2. Any polynomial $p \in I$ can be written as $p = h_1 \pi(e_1) + \ldots + h_m \pi(e_m)$.

Let $I = \langle f_1, \ldots, f_m \rangle$. The idea is to give each polynomial during the computations of the algorithm a so-called **signature**:

1. Let $e_1, \ldots, e_m \in R^m$ be canonical generators such that $\pi : R^m \to R$: $\pi(e_i) = f_i$ for all i.
2. Any polynomial $p \in I$ can be written as $p = h_1 \pi(e_1) + \ldots + h_m \pi(e_m)$.
3. Let k be the greatest index such that h_k is not zero. $\Rightarrow$ **A signature** $\mathcal{S}(p) = \mathrm{lm}(h_k) e_k$.

Signatures of polynomials

Let $I = \langle f_1, \ldots, f_m \rangle$. The idea is to give each polynomial during the computations of the algorithm a so-called **signature**:

1. Let $e_1, \ldots, e_m \in R^m$ be canonical generators such that $\pi : R^m \to R$: $\pi(e_i) = f_i$ for all i.
2. Any polynomial $p \in I$ can be written as $p = h_1 \pi(e_1) + \ldots + h_m \pi(e_m)$.
3. Let k be the greatest index such that h_k is not zero. $\Rightarrow$ **A signature** $\mathcal{S}(p) = \mathrm{lm}(h_k) e_k$.
4. A generating element f_i of I gets the signature $\mathcal{S}(f_i) = e_i$.

Let $I = \langle f_1, \dots, f_m \rangle$. The idea is to give each polynomial during the computations of the algorithm a so-called **signature**:

1. Let $e_1, \dots, e_m \in R^m$ be canonical generators such that $\pi : R^m \to R$: $\pi(e_i) = f_i$ for all i.
2. Any polynomial $p \in I$ can be written as $p = h_1\pi(e_1) + \ldots + h_m\pi(e_m)$.
3. Let k be the greatest index such that h_k is not zero. $\Rightarrow$ **A signature** $\mathcal{S}(p) = \mathrm{lm}(h_k)e_k$.
4. A generating element f_i of I gets the signature $\mathcal{S}(f_i) = e_i$.
5. Well-order $\prec$ on the set of all signatures $\Rightarrow$ Existence of **the minimal signature** of a polynomial p

Using **signatures** in a Gröbner basis algorithm we clearly need to define them **for s-polynomials**, too:

$$\mathrm{Spol}(p, q) = \mathrm{lc}(q) u_p p - \mathrm{lc}(p) u_q q$$

such that

$$\mathcal{S}\left(\mathrm{Spol}(p, q)\right) = \max\left\{u_p \mathcal{S}(p), u_q \mathcal{S}(q)\right\}$$

Computing Gröbner bases using signatures

Input: $G_{i-1} = \{g_1, \ldots, g_{r-1}\}$, a Gröbner basis of $\langle f_1, \ldots, f_{i-1} \rangle$
Output: Gröbner basis G of $\langle f_1, \ldots, f_i \rangle$

Computing Gröbner bases using signatures

Input: $G_{i-1} = \{g_1, \ldots, g_{r-1}\}$, a Gröbner basis of $\langle f_1, \ldots, f_{i-1} \rangle$
Output: Gröbner basis G of $\langle f_1, \ldots, f_i \rangle$

1. $g_r := f_i$

Computing Gröbner bases using signatures

Input: $G_{i-1} = \{g_1, \ldots, g_{r-1}\}$, a Gröbner basis of $\langle f_1, \ldots, f_{i-1} \rangle$
Output: Gröbner basis G of $\langle f_1, \ldots, f_i \rangle$

1. $g_r := f_i$
2. $G = \big\{(e_1, g_1), \ldots, (e_{r-1}, g_{r-1}), (e_r, g_r)\big\}$ (monic)

Computing Gröbner bases using signatures

Input: $G_{i-1} = \{g_1, \ldots, g_{r-1}\}$, a Gröbner basis of $\langle f_1, \ldots, f_{i-1} \rangle$
Output: Gröbner basis G of $\langle f_1, \ldots, f_i \rangle$

1. $g_r := f_i$
2. $G = \big\{(e_1, g_1), \ldots, (e_{r-1}, g_{r-1}), (e_r, g_r)\big\}$ (monic)
3. Set $P := \big\{s_{r,j}, g_r, g_j), j < r\big\}$

Computing Gröbner bases using signatures

Input: $G_{i-1} = \{g_1, \ldots, g_{r-1}\}$, a Gröbner basis of $\langle f_1, \ldots, f_{i-1} \rangle$
Output: Gröbner basis G of $\langle f_1, \ldots, f_i \rangle$

1. $g_r := f_i$
2. $G = \{(e_1, g_1), \ldots, (e_{r-1}, g_{r-1}), (e_r, g_r)\}$ (monic)
3. Set $P := \{s_{r,j}, g_r, g_j), j < r\}$
4. While $P \neq \emptyset$
   (a) Choose $(s, p, q) \in P$ such that s is minimal.
   (b) Delete (s, p, q) from P.

Computing Gröbner bases using signatures

Input: $G_{i-1} = \{g_1, \dots, g_{r-1}\}$, a Gröbner basis of $\langle f_1, \dots, f_{i-1} \rangle$
Output: Gröbner basis G of $\langle f_1, \dots, f_i \rangle$

1. $g_r := f_i$
2. $G = \{(e_1, g_1), \dots, (e_{r-1}, g_{r-1}), (e_r, g_r)\}$ (monic)
3. Set $P := \{s_{r,j}, g_r, g_j), j < r\}$
4. While $P \neq \emptyset$
   (a) Choose $(s, p, q) \in P$ such that s is minimal.
   (b) Delete (s, p, q) from P.
   (c) s not minimal for $up - vq \Rightarrow$ goto 4.

Computing Gröbner bases using signatures

Input: $G_{i-1} = \{g_1, \ldots, g_{r-1}\}$, a Gröbner basis of $\langle f_1, \ldots, f_{i-1} \rangle$
Output: Gröbner basis G of $\langle f_1, \ldots, f_i \rangle$

1. $g_r := f_i$
2. $G = \{(e_1, g_1), \ldots, (e_{r-1}, g_{r-1}), (e_r, g_r)\}$ (monic)
3. Set $P := \{s_{r,j}, g_r, g_j), j < r\}$
4. While $P \neq \emptyset$
   (a) Choose $(s, p, q) \in P$ such that s is minimal.
   (b) Delete (s, p, q) from P.
   (c) s not minimal for $up - vq \Rightarrow$ goto 4.
   (d) $(s, h) = \text{reduce}((s, up - vq), G)$

Computing Gröbner bases using signatures

Input: $G_{i-1} = \{g_1, \ldots, g_{r-1}\}$, a Gröbner basis of $\langle f_1, \ldots, f_{i-1} \rangle$
Output: Gröbner basis G of $\langle f_1, \ldots, f_i \rangle$

1. $g_r := f_i$
2. $G = \{(e_1, g_1), \ldots, (e_{r-1}, g_{r-1}), (e_r, g_r)\}$ (monic)
3. Set $P := \{s_{r,j}, g_r, g_j), j < r\}$
4. While $P \neq \emptyset$
   (a) Choose $(s, p, q) \in P$ such that s is minimal.
   (b) Delete (s, p, q) from P.
   (c) s not minimal for $up - vq \Rightarrow$ goto 4.
   (d) $(s, h) = \text{reduce}((s, up - vq), G)$
   (e) if $h \neq 0$ &
   $\nexists(\mathcal{S}(g), g) \in G,\ t \in M$ s.t. $t\mathcal{S}(g) = s$ and $t\,\mathrm{lm}(g) = \mathrm{lm}(h)$
      (i) For all $g \in G$ add $(s_{h,g}, h, g)$ to P.
      (ii) Add (s, h) to G.
5. When $P = \emptyset$ we are done and G is a Gröbner basis of $\langle f_1, \ldots, f_i \rangle$.

Computing Gröbner bases using signatures

Input: $G_{i-1} = \{g_1, \ldots, g_{r-1}\}$, a Gröbner basis of $\langle f_1, \ldots, f_{i-1} \rangle$
Output: Gröbner basis G of $\langle f_1, \ldots, f_i \rangle$

1. $g_r := f_i$
2. $G = \big\{(e_1, g_1), \ldots, (e_{r-1}, g_{r-1}), (e_r, g_r)\big\}$ (monic)
3. Set $P := \big\{s_{r,j}, g_r, g_j), j < r\big\}$
4. While $P \neq \emptyset$
   (a) Choose $(s, p, q) \in P$ such that s is minimal.
   (b) Delete (s, p, q) from P.
   (c) s not minimal for $up - vq \Rightarrow$ goto 4.
   (d) $\big(s, h\big) = \text{reduce}\big((s, up - vq), G\big) \Leftarrow$ sig-safe!
   (e) if $h \neq 0$ &
   $\nexists(\mathcal{S}(g), g) \in G,\ t \in M$ s.t. $t\mathcal{S}(g) = s$ and $t\,\mathrm{lm}(g) = \mathrm{lm}(h)$
      (i) For all $g \in G$ add $\big(s_{h,g}, h, g\big)$ to P.
      (ii) Add $\big(s, h\big)$ to G.
5. When $P = \emptyset$ we are done and G is a Gröbner basis of $\langle f_1, \ldots, f_i \rangle$.

Let $\big(\mathcal{S}(p), p\big)$, $\big(\mathcal{S}(q), q\big)$ such that $\lambda \mathrm{lm}(q) = \mathrm{lm}(p)$.

Let $\big(\mathcal{S}(p), p\big)$, $\big(\mathcal{S}(q), q\big)$ such that $\lambda \mathrm{lm}(q) = \mathrm{lm}(p)$.

1. **Sig-safe:** $\mathcal{S}(p - \lambda q) = \mathcal{S}(p) \Rightarrow \mathcal{S}(p) \succ \lambda \mathcal{S}(q)$.
2. **Sig-unsafe:** $\mathcal{S}(p - \lambda q) = \lambda \mathcal{S}(q) \Rightarrow \mathcal{S}(p) \prec \lambda \mathcal{S}(q)$.
3. **Sig-cancelling:** $\mathcal{S}(p) = \lambda \mathcal{S}(q) \Rightarrow \mathcal{S}(p - \lambda q) =?$

Termination?

1. No new s-polynomials for $\big(\mathcal{S}(h), h\big) = \lambda\big(\mathcal{S}(g), g\big)$
2. Each new element expands $\big\langle\big(\mathcal{S}(h), \mathrm{lm}(h)\big)\big\rangle$

Computing Gröbner bases using signatures

Termination?

1. No new s-polynomials for $\big(\mathcal{S}(h), h\big) = \lambda\big(\mathcal{S}(g), g\big)$
2. Each new element expands $\big\langle\big(\mathcal{S}(h), \mathrm{lm}(h)\big)\big\rangle$

Correctness?

1. Proceed by minimal signature in P
2. All s-polynomials considered:
   sig-unsafe reduction $\Rightarrow$ new critical pair next round
3. All nonzero elements added besides $\big(\mathcal{S}(h), h\big) = \lambda\big(\mathcal{S}(g), g\big)$

Allowed criteria?

Non-minimal signature (NM)
$\mathcal{S}(h)$ not minimal for h? $\Rightarrow$ discard h

Non-minimal signature (NM)

$\mathcal{S}(h)$ not minimal for h? $\Rightarrow$ discard h

Proof.

1. There exists syzygy s with $\mathrm{lm}(s) = \mathcal{S}(h)$.
2. We can rewrite h using a lower signature.
3. We proceed by increasing signatures.
   $\Rightarrow$ Those reductions are already considered.

□

Rewritable signature (RW)
$\mathcal{S}(g) = \mathcal{S}(h)$? $\Rightarrow$ discard either g or h

Rewritable signature (RW)
$\mathcal{S}(g) = \mathcal{S}(h)$? $\Rightarrow$ discard either g or h

Proof.

1. $\mathcal{S}(g - h) \prec \mathcal{S}(h), \mathcal{S}(g)$.
2. We proceed by increasing signatures.
   $\Rightarrow$ Those reductions are already considered.
   $\Rightarrow$ We can rewrite $h = g +$ terms of lower signature.

□

The following section is about

$$H = \left\{ \mathrm{lm}(g_1), \ldots, \mathrm{lm}(g_{r-1}) \right\}.$$

$H = \left\{\mathrm{lm}(g_1), \ldots, \mathrm{lm}(g_{r-1})\right\}$.
If

$$\mathcal{S}(g) = \sigma e_r, \exists h \in H \text{ such that } h \mid \sigma,$$

then discard g.
(There exists a principal syzygy $g_i e_r - g_r e_i, h = \mathrm{lm}(g_i), i < r$.)

$H = \{\mathrm{lm}(g_1), \ldots, \mathrm{lm}(g_{r-1})\}$.
If

$$\mathcal{S}(g) = \sigma e_r, \exists h \in H \text{ such that } h \mid \sigma,$$

then discard g.
(There exists a principal syzygy $g_i e_r - g_r e_i, h = \mathrm{lm}(g_i), i < r$.)

Only in G2V: Whenever p reduces to zero

$$\Rightarrow H = H \cup \{\lambda\} \text{ where } \mathcal{S}(p) = \lambda e_r.$$

Implementation of (RW)

Quite different in F5 and G2V:

1. F5 implements (RW) **very aggressive** using divisibility instead of equality.
2. G2V just uses the **generic and soft** (RW) when adding new critical pairs to the pair set.

F5E – Combine the ideas

Behaviour depending on number of zero reductions

- G2V actively uses zero reductions to improve (NM).
- F5 does not do this, but possible incorporates some of this data in (RW).
- Checking by F5's (RW) costs much more time than checking by (NM).

Differences in the reduction process

Remark

The presented criteria (NM) and (RW) are also used during the (sig-safe) reduction steps. This usage is quite **soft in G2V** and quite **aggressive in F5**.

$\Rightarrow$ **Termination:** G2V ☺ – F5 ☹

The following section is about

Number of critical pairs and zero reductions

System	F5		F5E		G2V	
Katsura 9	886	0	886	0	886	0
Katsura 10	1,781	0	1,781	0	1,781	0
Eco 8	830	322	**565**	**57**	2,012	**57**
Eco 9	2,087	929	**1,278**	**120**	5,794	**120**
F744	1,324	342	**1,151**	**169**	2,145	**169**
Cyclic 7	1,018	76	**978**	**36**	3,072	**36**
Cyclic 8	7,066	244	**5,770**	**244**	24,600	**244**

Timings in seconds

System	F5	F5E	G2V
Katsura 9	14.98	**14.87**	17.63
Katsura 10	153.35	**152.39**	192.20
Eco 8	2.24	**0.38**	0.49
Eco 9	77.13	**8.19**	13.51
F744	19.35	**8.79**	26.86
Cyclic 7	**7.01**	7.22	33.85
Cyclic 8	7,310.39	**4,961.58**	26,242.12

The following section is about

Outlook

- **Efficient open source implementation:**
  Ongoing task, part of SINGULAR's restructuring
- **Parallelization:**
  On criteria checks, needs thread-safe memory management
- **Syzygy computations:**
  Needs implementation
- **Signature orders:**
  Non-incremental for non-complete intersections?

References

[AH09] G. Ars and A. Hashemi. Extended F5 Criteria

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

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

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

[Fa02] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero F_5

[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 Gröbner Bases

[SIN11] W. Decker, G.-M. Greuel, G. Pfister and H. Schönemann. SINGULAR 3-1-3. *A computer algebra system for polynomial computations*, University of Kaiserslautern, 2011, `http://www.singular.uni-kl.de`.

[SW10] Y. Sun and D. Wang. A new proof of the F5 Algorithm

[SW11] Y. Sun and D. Wang. A Generalized Criterion for Signature Related Gröbner Basis Algorithms