Extensions of signature Gröbner bases: rings and the free algebra

Thibaut Verron

Seminar “Algebra and Discrete Mathematics”

Johannes Kepler University

21 October 2021

Gröbner bases

Gröbner bases for commutative polynomials over fields:

- solving equations (parametrization, elimination, dimension of the solutions...)
- simplifications, reductions, computations in modules
- **with signatures**: optimization, computation of syzygies and cofactors

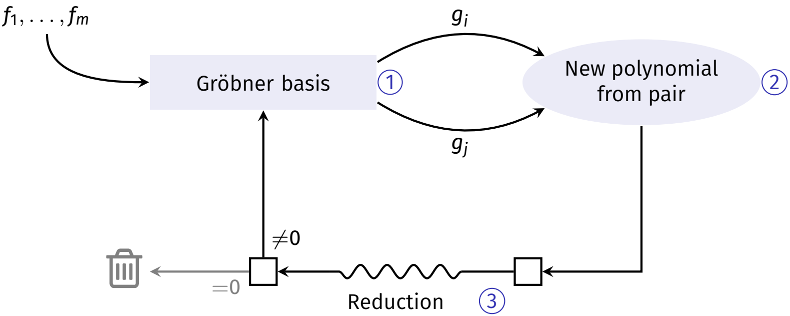
This talk: two generalizations of signatures

- Gröbner bases over $\mathbb{Z}$
- Gröbner bases on the free algebra

Notations:

- R ring or field
- Commutative polynomial algebra: $A = R[X_1, \ldots, X_n]$ with a monomial order $<$
- Commutative monomial: $\mathbf{X}^{\mathbf{a}} = X_1^{a_1} \cdots X_n^{a_n}$
- Free algebra: $A = R\langle X_1, \ldots, X_n \rangle$ with a monomial order $<$
- Noncommutative monomial (word): $X_{i_1} X_{i_2} \cdots X_{i_d}$

$$f = \overbrace{\underbrace{c}_{\mathrm{lc}(f)} \cdot \underbrace{\mathbf{X}^{\mathbf{a}}}_{\mathrm{lm}(f)}}^{\mathrm{lt}(f)} + \text{ smaller terms}$$



1. **Selection**: different strategies
2. **Construction**: S-polynomials: $\text{S-Pol}(g_i, g_j) = \dfrac{\text{lcmlt}(g_i, g_j)}{\text{lt}(g_i)} g_i - \dfrac{\text{lcmlt}(g_i, g_j)}{\text{lt}(g_j)} g_j$
3. **Reduction**: if $\text{lt}(f) = t\text{lt}(g)$, $f \to f - tg$

Reminder on signature Gröbner basis algorithms

Setting:

- Given $f_1, \ldots, f_m \in A = R[\mathbf{X}]$ generating the ideal I
- A-module $A^m = A\mathbf{e}_1 \oplus \cdots \oplus A\mathbf{e}_m$ with a A-morphism $\pi : A^m \to I$, $\mathbf{e}_i \mapsto f_i$
- A-module $\mathcal{I} = \{(\mathbf{f}, \pi(\mathbf{f})) : \mathbf{p} \in A^m\} \subseteq A^m \times I$
- $\mathcal{I}$ is isomorphic to A^m, we use the same notation: if $f = \pi(f)$, $\mathbf{f} \equiv (\mathbf{f}, f) \equiv f^{[\mathbf{f}]}$

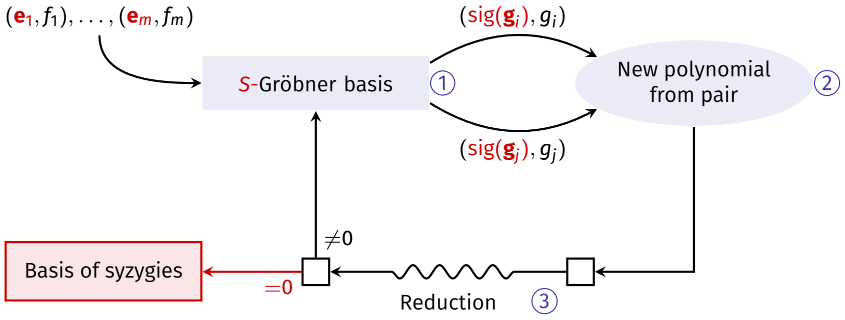
Signatures:

- Assign a monomial ordering on A^m (compatible with that on A)
- Signature of $\mathbf{f}$: sig($\mathbf{f}$) = leading monomial of $\mathbf{f} \in A^m$ for that ordering
- We use sig for the leading monomial of the **module part**
- We keep using lt, etc. for the leading term of the **polynomial part**: $\mathrm{lt}(\mathbf{f}) = \mathrm{lt}(f)$

Regular operations

- If $\mathrm{sig}(\mathbf{f}) > \mathrm{sig}(\mathbf{g})$, $\mathbf{f} - \mathbf{g}$ is a regular operation (the signature is preserved)
- If $\mathrm{sig}(\mathbf{f}) = \mathrm{sig}(\mathbf{g})$, $\mathbf{f} - \mathbf{g}$ is a singular operation (the signature may drop)

Buchberger's algorithm, with signatures (*R* is a field)



1. **Selection**: non-decreasing signatures
2. **Construction**: regular S-polynomials: $\text{S-Pol}(\mathbf{g}_i, \mathbf{g}_j) = \dfrac{\mathrm{lcmlt}(\mathbf{g}_i, \mathbf{g}_j)}{\mathrm{lt}(\mathbf{g}_i)}\mathbf{g}_i - \dfrac{\mathrm{lcmlt}(\mathbf{g}_i, \mathbf{g}_j)}{\mathrm{lt}(\mathbf{g}_j)}\mathbf{g}_j$
3. **Reduction**: regular s-reductions: if $\mathrm{lt}(\mathbf{f}) = t\,\mathrm{lt}(\mathbf{g})$ and $t\,\mathrm{sig}(\mathbf{g}) \lneq \mathrm{sig}(\mathbf{f})$, $\mathbf{f} \to \mathbf{f} - t\mathbf{g}$

Part 1: signature Gröbner bases over $\mathbb{Z}$

Joint work with Maria Francis

(Indian Institute of Technology Hyderabad)

Summary of Gröbner basis algorithms over rings

Two questions:

- How to compute S-polynomials?
- How to compute reductions?

Field	Buchberger (1965) Faugère: F4 (1999) ⋮	Usual // Usual Usual // Usual (linear algebra)

Summary of Gröbner basis algorithms over rings

Two questions:

- How to compute S-polynomials?
- How to compute reductions?

Field	Buchberger (1965)	Usual // Usual
	Faugère: F4 (1999)	Usual // Usual (linear algebra)
	$\vdots$	
General (Noetherian) ring	Möller weak (1988)	Multiple // Multiple

Summary of Gröbner basis algorithms over rings

Two questions:

- How to compute S-polynomials?
- How to compute reductions?

Ring	Algorithm	S-polynomials // Reductions
Field	Buchberger (1965)	Usual // Usual
	Faugère: F4 (1999)	Usual // Usual (linear algebra)
	⋮	
Principal ideal domain	Möller strong (1988)	Usual // Usual with G-pol
	Pan (1989)	Usual or G-pols // Usual
General (Noetherian) ring	Möller weak (1988)	Multiple // Multiple

Summary of Gröbner basis algorithms over rings

Two questions:

- How to compute S-polynomials?
- How to compute reductions?

Ring	Algorithm	S-polynomials // Reductions
Field	Buchberger (1965)	Usual // Usual
	Faugère: F4 (1999)	Usual // Usual (linear algebra)
	⋮	
Euclidean ring	Kandri-Rody, Kapur (1988)	Usual and G-pols // Usual
	Lichtblau (2012)	Usual or G-pols // Usual
Principal ideal domain	Möller strong (1988)	Usual // Usual with G-pol
	Kandri-Rody, Kapur (1988)	Usual and G-pols // Usual
	Pan (1989)	Usual or G-pols // Usual
General (Noetherian) ring	Möller weak (1988)	Multiple // Multiple

Signatures over $\mathbb{Z}$

Definition

- Over fields: signature of $\mathbf{f}$ = leading monomial of the module part of $\mathbf{f}$

  = monomial $m\mathbf{e}_i$ in A^m such that $f = cmf_i +$ "smaller" terms
- In that case, c does not matter!
- Over rings, we cannot divide by c and we need to keep the coefficient in the signature
- The signature of f is $cm\mathbf{e}_i$

Consequence for operations

- If $\mathrm{sig}(\mathbf{f}) > \mathrm{sig}(\mathbf{g})$, $\mathbf{f} - \mathbf{g}$ is a regular operation (the signature is preserved)
- If $\mathrm{sig}(\mathbf{f}) = \mathrm{sig}(\mathbf{g})$, $\mathbf{f} - \mathbf{g}$ is a singular operation (the signature does drop)
- If $\mathrm{sig}(\mathbf{f}) \simeq \mathrm{sig}(\mathbf{g})$ with different coefficients, $\mathbf{f} - \mathbf{g}$ has signature $\mathrm{sig}(\mathbf{f}) - \mathrm{sig}(\mathbf{g})$

Main question: how to order the signatures with their coefficients?

Summary of Gröbner basis algorithms over rings with signatures

Three questions:

- How to compute S-polynomials?
- How to compute reductions?
- How to order signatures?

Case of fields: partial order is enough

Field	Buchberger (1965) → B. with sig.	Usual // Usual
	Faugère: F4 (1999) → F5 (2002)	Usual // Usual (linear algebra)
	⋮	
Euclidean ring	Kandri-Rody, Kapur (1988)	Usual and G-pols // Usual
	Lichtblau (2012)	Usual or G-pols // Usual
Principal ideal domain	Möller strong (1988)	Usual // Usual with G-pol
	Kandri-Rody, Kapur (1988)	Usual and G-pols // Usual
	Pan (1989)	Usual or G-pols // Usual
General (Noetherian) ring	Möller weak (1988)	Multiple // Multiple

Summary of Gröbner basis algorithms over rings with signatures

Three questions:

- How to compute S-polynomials?
- How to compute reductions?
- How to order signatures?

Case of fields: partial order is enough
[Eder, Pfister, Popescu 2017]: cannot order coefs

Ring	Algorithms	S-polynomials // Reductions
Field	Buchberger (1965) $\to$ B. with sig.	Usual // Usual
	Faugère: F4 (1999) $\to$ F5 (2002)	Usual // Usual (linear algebra)
	⋮	
Euclidean ring	Kandri-Rody, Kapur (1988)	Usual and G-pols // Usual
	Lichtblau (2012)	Usual or G-pols // Usual
Principal ideal domain	Möller strong (1988)	Usual // Usual with G-pol
	Kandri-Rody, Kapur (1988)	Usual and G-pols // Usual
	Pan (1989)	Usual or G-pols // Usual
General (Noetherian) ring	Möller weak (1988)	Multiple // Multiple

Summary of Gröbner basis algorithms over rings with signatures

Three questions:

- How to compute S-polynomials?
- How to compute reductions?
- How to order signatures?

Case of fields: partial order is enough
[Eder, Pfister, Popescu 2017]: cannot order coefs
[Francis, V. 2018]: partial order is enough

Ring	Algorithms	S-polynomials // Reductions
Field	Buchberger (1965) $\rightarrow$ B. with sig.	Usual // Usual
	Faugère: F4 (1999) $\rightarrow$ F5 (2002)	Usual // Usual (linear algebra)
	⋮	
Euclidean ring	Kandri-Rody, Kapur (1988)	Usual and G-pols // Usual
	Lichtblau (2012)	Usual or G-pols // Usual
Principal ideal domain	Möller weak with sig (2018)	
	Möller strong (1988)$\rightarrow$ with sig (2019)	Usual // Usual with G-pol
	Kandri-Rody, Kapur (1988)	Usual and G-pols // Usual
	Pan (1989)	Usual or G-pols // Usual
General (Noetherian) ring	Möller weak (1988)	Multiple // Multiple

Summary of Gröbner basis algorithms over rings with signatures

Three questions:

- How to compute S-polynomials?
- How to compute reductions?
- How to order signatures?

Case of fields: partial order is enough
[Eder, Pfister, Popescu 2017]: cannot order coefs
[Francis, V. 2018]: partial order is enough

Ring	Algorithms	S-polynomials // Reductions
Field	Buchberger (1965) $\rightarrow$ B. with sig.	Usual // Usual
	Faugère: F4 (1999) $\rightarrow$ F5 (2002)	Usual // Usual (linear algebra)
	$\vdots$	
Euclidean ring	Kandri-Rody, Kapur (1988)	Usual and G-pols // Usual
	Lichtblau (2012)	Usual or G-pols // Usual
Principal ideal domain	Möller weak with sig (2018)	
	Möller strong (1988) $\rightarrow$ with sig (2019)	Usual // Usual with G-pol
	Kandri-Rody, Kapur (1988)	Usual and G-pols // Usual
	Pan (1989)	Usual or G-pols // Usual
General (Noetherian) ring	Möller weak (1988)	Multiple // Multiple

This work: signature variants of the algos of Kandri-Rody and Kapur, and of Pan/Lichtblau

What are G-polynomials?

Example: $f = 3x$, $g = 2y$, $I = \langle f, g \rangle$

- Not a strong Gröbner basis: $xy = yf - xg \in I$ is not reducible by f or g
- Adding $\text{S-Pol}(f, g) = 0$ does not help
- $\text{G-Pol}(f, g) = xy$

What are G-polynomials?

Example: $f = 3x$, $g = 2y$, $I = \langle f, g \rangle$

- Not a strong Gröbner basis: $xy = yf - xg \in I$ is not reducible by f or g
- Adding S-Pol$(f, g) = 0$ does not help
- G-Pol$(f, g) = xy$

Definition

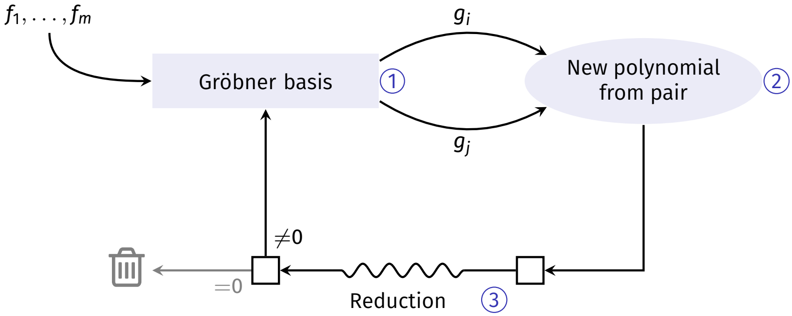
$\mathbf{f}, \mathbf{g} \in \mathcal{I}$, u, v Bézout coefficients for $\mathrm{lc}(\mathbf{f}), \mathrm{lc}(\mathbf{g})$

- $\text{G-Pol}(\mathbf{f}, \mathbf{g}) = u \dfrac{\mathrm{lcmlm}(\mathbf{f}, \mathbf{g})}{\mathrm{lm}(\mathbf{f})} \mathbf{f} + v \dfrac{\mathrm{lcmlm}(\mathbf{f}, \mathbf{g})}{\mathrm{lm}(\mathbf{g})} \mathbf{g}$

Main properties

- $\mathrm{lc}(\text{G-Pol}(\mathbf{f}, \mathbf{g})) = \mathrm{gcdlc}(\mathbf{f}, \mathbf{g})$
- If $\mathrm{lt}(\mathbf{f}) = t_1 \mathrm{lt}(\mathbf{g}_1) + t_2 \mathrm{lt}(\mathbf{g}_2)$, then $\mathbf{f}$ is reducible by G-Pol$(\mathbf{g}_1, \mathbf{g}_2)$
- One can always choose u, v such that

$$\mathrm{sig}(\text{G-Pol}(\mathbf{f}, \mathbf{g})) \simeq \max\left(\frac{\mathrm{lcmlm}(\mathbf{f}, \mathbf{g})}{\mathrm{lm}(\mathbf{f})} \mathrm{sig}(\mathbf{f}), \frac{\mathrm{lcmlm}(\mathbf{f}, \mathbf{g})}{\mathrm{lm}(\mathbf{g})} \mathrm{sig}(\mathbf{g})\right)$$



1. **Selection**: different strategies
2. **Construction**: S-polynomial

   and G-polynomial if $\mathrm{lc}(g_i)$ and $\mathrm{lc}(g_j)$ do not divide each other
3. **Reduction**

G-polynomials for syzygies

Need a similar construction to capture all possible combinations of syzygy signatures.

Definition

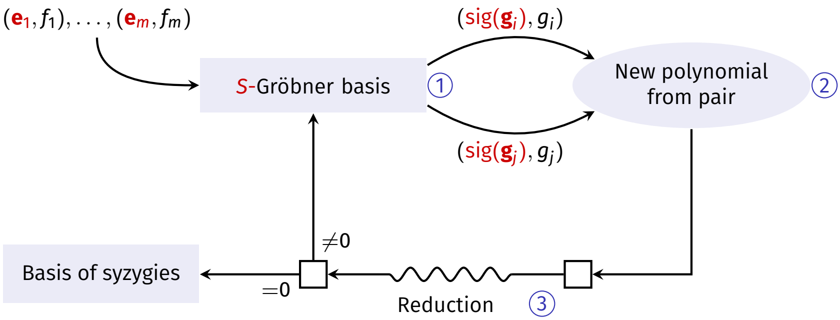
$\mathbf{z}_1, \mathbf{z}_2 \in \mathrm{Syz}(\mathcal{I})$ with $\mathrm{sig}(\mathbf{z}_i) = a_i m_i \mathbf{e}_j$; u, v Bézout coefficients for a_1, a_2

- $\text{G-Pol}(\mathbf{z}_1, \mathbf{z}_2) = u \dfrac{\mathrm{lcm}(m_1, m_2)}{m_1} \mathbf{z}_1 + v \dfrac{\mathrm{lcm}(m_1, m_2)}{m_2} \mathbf{z}_2$

Main properties

- $\mathrm{sig}(\text{G-Pol}(\mathbf{z}_1, \mathbf{z}_2)) = \gcd(a_1, a_2) \mathrm{lcm}(m_1, m_2) \mathbf{e}_j$
- If $\mathrm{sig}(\mathbf{f}) = t_1 \mathrm{sig}(\mathbf{z}_1) + t_2 \mathrm{sig}(\mathbf{z}_2)$, then $\mathbf{f}$ is sig-reducible by $\text{G-Pol}(\mathbf{z}_1, \mathbf{z}_2)$
- No need to be careful about the choice of u, v

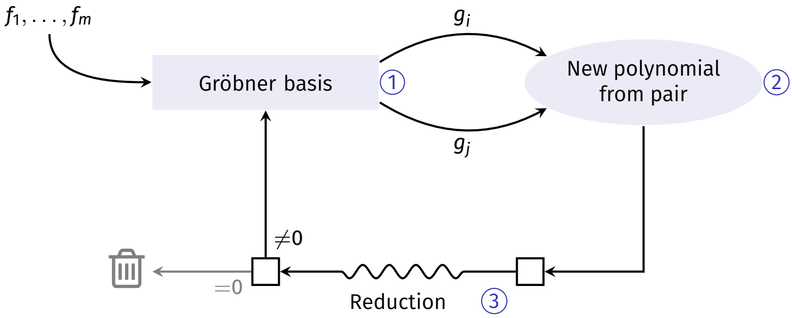
Kandri-Rody and Kapur's algorithm, with signatures (R is a PID)



1. **Selection**: non-decreasing signatures
2. **Construction**: regular S-polynomial

   and G-polynomial if $\mathrm{lc}(\mathbf{g}_i)$ and $\mathrm{lc}(\mathbf{g}_j)$ do not divide each other
3. **Reduction**: regular

Pan/Lichtblau's algorithm (R is a PID)



1. **Selection**: different strategies
2. **Construction**: S-polynomial if one of $\mathrm{lc}(g_i)$ and $\mathrm{lc}(g_j)$ divides the other

   or G-polynomial if $\mathrm{lc}(g_i)$ and $\mathrm{lc}(g_j)$ do not divide each other
3. **Reduction**

Why does it work?

Idea:

- Let f and g with $a = \mathrm{lc}(f)$ and $b = \mathrm{lc}(g)$ not dividing each other, let $d = \mathrm{gcdlc}(f, g)$
- How to recover $\text{S-Pol}(f, g) = \dfrac{b}{d}\mu f - \dfrac{a}{d}\nu g$?

Why does it work?

Idea:

- Let f and g with $a = \mathrm{lc}(f)$ and $b = \mathrm{lc}(g)$ not dividing each other, let $d = \mathrm{gcdlc}(f, g)$
- How to recover $\text{S-Pol}(f, g) = \frac{b}{d}\mu f - \frac{a}{d}\nu g$?
- The algorithm computes $h = \text{G-Pol}(f, g) = u\mu f + v\nu g$, with $\mathrm{lc}(h) = d$

Why does it work?

Idea:

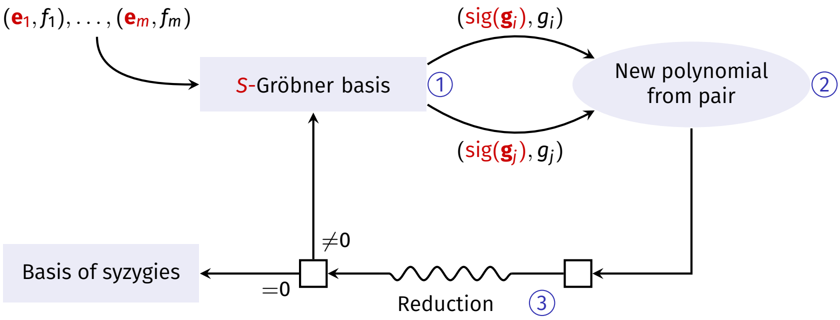
- Let f and g with $a = \operatorname{lc}(f)$ and $b = \operatorname{lc}(g)$ not dividing each other, let $d = \operatorname{gcdlc}(f,g)$
- How to recover $\text{S-Pol}(f,g) = \frac{b}{d}\mu f - \frac{a}{d}\nu g$?

- The algorithm computes $h = \text{G-Pol}(f,g) = u\mu f + v\nu g$, with $\operatorname{lc}(h) = d$
- $\operatorname{lc}(h)$ divides both $\operatorname{lc}(f)$ and $\operatorname{lc}(g)$, and the algorithm computes the S-polynomials:

$$\begin{aligned} \text{S-Pol}(f,h) &= \mu f - \frac{a}{d}h \\ &= \left(1 - \frac{ua}{d}\right)\mu f - \frac{av}{d}\mu g \\ &= \frac{vb}{d}\mu f - \frac{av}{d}\nu g \\ &= v\text{S-Pol}(f,g) \end{aligned}$$

$$\text{S-Pol}(g,h) = u\text{S-Pol}(f,g)$$

Pan/Lichtblau's algorithm, with signatures (R is a PID)



1. **Selection**: non-decreasing signatures
2. **Construction**: non-singular S-polynomial if one of $\mathrm{lc}(\mathbf{g}_i)$ and $\mathrm{lc}(\mathbf{g}_j)$ divides the other

   or G-polynomial if $\mathrm{lc}(\mathbf{g}_i)$ and $\mathrm{lc}(\mathbf{g}_j)$ do not divide each other
3. **Reduction**: regular

Why does it work?

Idea:

- Let $\mathbf{f}$ and $\mathbf{g}$ with $a = \mathrm{lc}(\mathbf{f})$ and $b = \mathrm{lc}(\mathbf{g})$ not dividing each other, let $d = \mathrm{gcdlc}(\mathbf{f}, \mathbf{g})$
- How to recover $\text{S-Pol}(\mathbf{f}, \mathbf{g}) = \frac{b}{d}\mu\mathbf{f} - \frac{a}{d}\nu\mathbf{g}$?

- The algorithm computes $\mathbf{h} = \text{G-Pol}(\mathbf{f}, \mathbf{g}) = u\mu\mathbf{f} + v\nu\mathbf{g}$, with $\mathrm{lc}(\mathbf{h}) = d$
- $\mathrm{lc}(\mathbf{h})$ divides both $\mathrm{lc}(\mathbf{f})$ and $\mathrm{lc}(\mathbf{g})$, and the algorithm computes the S-polynomials:

$$\begin{aligned}
\text{S-Pol}(\mathbf{f}, \mathbf{h}) &= \mu\mathbf{f} - \frac{a}{d}\mathbf{h} \\
&= \left(1 - \frac{ua}{d}\right)\mu\mathbf{f} - \frac{av}{d}\mu\mathbf{g} \\
&= \frac{vb}{d}\mu\mathbf{f} - \frac{av}{d}\nu\mathbf{g} \\
&= v\text{S-Pol}(\mathbf{f}, \mathbf{g})
\end{aligned}$$

$$\text{S-Pol}(\mathbf{g}, \mathbf{h}) = u\text{S-Pol}(\mathbf{f}, \mathbf{g})$$

Why does it work?

Idea:

sig. $\mathbf{s}$ $\mathbf{t}$ with $\mu\mathbf{s} \gneq \nu\mathbf{t}$

- Let $\mathbf{f}$ and $\mathbf{g}$ with $a = \mathrm{lc}(\mathbf{f})$ and $b = \mathrm{lc}(\mathbf{g})$ not dividing each other, let $d = \mathrm{gcdlc}(\mathbf{f}, \mathbf{g})$
- How to recover $\text{S-Pol}(\mathbf{f}, \mathbf{g}) = \frac{b}{d}\mu\mathbf{f} - \frac{a}{d}\nu\mathbf{g}$?
- The algorithm computes $\mathbf{h} = \text{G-Pol}(\mathbf{f}, \mathbf{g}) = u\mu\mathbf{f} + v\nu\mathbf{g}$, with $\mathrm{lc}(\mathbf{h}) = d$
- $\mathrm{lc}(\mathbf{h})$ divides both $\mathrm{lc}(\mathbf{f})$ and $\mathrm{lc}(\mathbf{g})$, and the algorithm computes the S-polynomials:

$$\begin{aligned}
\text{S-Pol}(\mathbf{f}, \mathbf{h}) &= \mu\mathbf{f} - \frac{a}{d}\mathbf{h} \\
&= \left(1 - \frac{ua}{d}\right)\mu\mathbf{f} - \frac{av}{d}\mu\mathbf{g} \\
&= \frac{vb}{d}\mu\mathbf{f} - \frac{av}{d}\nu\mathbf{g} \\
&= v\text{S-Pol}(\mathbf{f}, \mathbf{g})
\end{aligned}$$

$$\text{S-Pol}(\mathbf{g}, \mathbf{h}) = u\text{S-Pol}(\mathbf{f}, \mathbf{g})$$

Why does it work?

Idea:

sig. $\mathbf{s}$ $\mathbf{t}$ with $\mu\mathbf{s} \gneq \nu\mathbf{t}$

- Let $\mathbf{f}$ and $\mathbf{g}$ with $a = \text{lc}(\mathbf{f})$ and $b = \text{lc}(\mathbf{g})$ not dividing each other, let $d = \text{gcdlc}(\mathbf{f}, \mathbf{g})$
- How to recover $\text{S-Pol}(\mathbf{f}, \mathbf{g}) = \boxed{\frac{b}{d}\mu\mathbf{f} - \frac{a}{d}\nu\mathbf{g}}$? Regular, sig. $\simeq \mu\mathbf{s}$
- The algorithm computes $\mathbf{h} = \text{G-Pol}(\mathbf{f}, \mathbf{g}) = u\mu\mathbf{f} + v\nu\mathbf{g}$, with $\text{lc}(\mathbf{h}) = d$
- $\text{lc}(\mathbf{h})$ divides both $\text{lc}(\mathbf{f})$ and $\text{lc}(\mathbf{g})$, and the algorithm computes the S-polynomials:

$$\begin{aligned}\text{S-Pol}(\mathbf{f}, \mathbf{h}) &= \mu\mathbf{f} - \frac{a}{d}\mathbf{h} \\ &= \left(1 - \frac{ua}{d}\right)\mu\mathbf{f} - \frac{av}{d}\mu\mathbf{g} \\ &= \frac{vb}{d}\mu\mathbf{f} - \frac{av}{d}\nu\mathbf{g} \\ &= v\text{S-Pol}(\mathbf{f}, \mathbf{g})\end{aligned}$$

$$\text{S-Pol}(\mathbf{g}, \mathbf{h}) = u\text{S-Pol}(\mathbf{f}, \mathbf{g})$$

Why does it work?

Idea:

sig. **s** **t** with $\mu\mathbf{s} \gneqq \nu\mathbf{t}$

- Let $\mathbf{f}$ and $\mathbf{g}$ with $a = \mathrm{lc}(\mathbf{f})$ and $b = \mathrm{lc}(\mathbf{g})$ not dividing each other, let $d = \mathrm{gcdlc}(\mathbf{f}, \mathbf{g})$
- How to recover $\text{S-Pol}(\mathbf{f}, \mathbf{g}) = \frac{b}{d}\mu\mathbf{f} - \frac{a}{d}\nu\mathbf{g}$? Regular, sig. $\simeq \mu\mathbf{s}$

  sig. $\simeq \mu\mathbf{s}$
- The algorithm computes $\mathbf{h} = \text{G-Pol}(\mathbf{f}, \mathbf{g}) = u\mu\mathbf{f} + v\nu\mathbf{g}$, with $\mathrm{lc}(\mathbf{h}) = d$
- $\mathrm{lc}(\mathbf{h})$ divides both $\mathrm{lc}(\mathbf{f})$ and $\mathrm{lc}(\mathbf{g})$, and the algorithm computes the S-polynomials:

$$\begin{aligned}
\text{S-Pol}(\mathbf{f}, \mathbf{h}) &= \mu\mathbf{f} - \frac{a}{d}\mathbf{h} \\
&= \left(1 - \frac{ua}{d}\right)\mu\mathbf{f} - \frac{av}{d}\mu\mathbf{g} \\
&= \frac{vb}{d}\mu\mathbf{f} - \frac{av}{d}\nu\mathbf{g} \\
&= v\text{S-Pol}(\mathbf{f}, \mathbf{g})
\end{aligned}$$

$$\text{S-Pol}(\mathbf{g}, \mathbf{h}) = u\text{S-Pol}(\mathbf{f}, \mathbf{g})$$

Why does it work?

Idea:

sig. **s** **t** with $\mu\mathbf{s} \gneq \nu\mathbf{t}$

- Let **f** and **g** with $a = \mathrm{lc}(\mathbf{f})$ and $b = \mathrm{lc}(\mathbf{g})$ not dividing each other, let $d = \mathrm{gcdlc}(\mathbf{f}, \mathbf{g})$
- How to recover $\text{S-Pol}(\mathbf{f}, \mathbf{g}) = \frac{b}{d}\mu\mathbf{f} - \frac{a}{d}\nu\mathbf{g}$? Regular, sig. $\simeq \mu\mathbf{s}$

sig. $\simeq \mu\mathbf{s}$

- The algorithm computes $\mathbf{h} = \text{G-Pol}(\mathbf{f}, \mathbf{g}) = u\mu\mathbf{f} + v\nu\mathbf{g}$, with $\mathrm{lc}(\mathbf{h}) = d$
- $\mathrm{lc}(\mathbf{h})$ divides both $\mathrm{lc}(\mathbf{f})$ and $\mathrm{lc}(\mathbf{g})$, and the algorithm computes the S-polynomials:

$\simeq \mu\mathbf{s}$ $\simeq \mu\mathbf{s}$ not regular

$$\begin{aligned}
\text{S-Pol}(\mathbf{f}, \mathbf{h}) &= \mu\mathbf{f} - \frac{a}{d}\mathbf{h} \\
&= \left(1 - \frac{ua}{d}\right)\mu\mathbf{f} - \frac{av}{d}\mu\mathbf{g} \\
&= \frac{vb}{d}\mu\mathbf{f} - \frac{av}{d}\nu\mathbf{g} \\
&= v\,\text{S-Pol}(\mathbf{f}, \mathbf{g})
\end{aligned}$$

$$\text{S-Pol}(\mathbf{g}, \mathbf{h}) = u\,\text{S-Pol}(\mathbf{f}, \mathbf{g})$$

Why does it work?

Idea:

sig. $\mathbf{s}$ $\mathbf{t}$ with $\mu\mathbf{s} \gneq \nu\mathbf{t}$

- Let $\mathbf{f}$ and $\mathbf{g}$ with $a = \mathrm{lc}(\mathbf{f})$ and $b = \mathrm{lc}(\mathbf{g})$ not dividing each other, let $d = \mathrm{gcdlc}(\mathbf{f}, \mathbf{g})$
- How to recover $\text{S-Pol}(\mathbf{f}, \mathbf{g}) = \frac{b}{d}\mu\mathbf{f} - \frac{a}{d}\nu\mathbf{g}$? Regular, sig. $\simeq \mu\mathbf{s}$

  sig. $\simeq \mu\mathbf{s}$
- The algorithm computes $\mathbf{h} = \text{G-Pol}(\mathbf{f}, \mathbf{g}) = u\mu\mathbf{f} + v\nu\mathbf{g}$, with $\mathrm{lc}(\mathbf{h}) = d$
- $\mathrm{lc}(\mathbf{h})$ divides both $\mathrm{lc}(\mathbf{f})$ and $\mathrm{lc}(\mathbf{g})$, and the algorithm computes the S-polynomials:

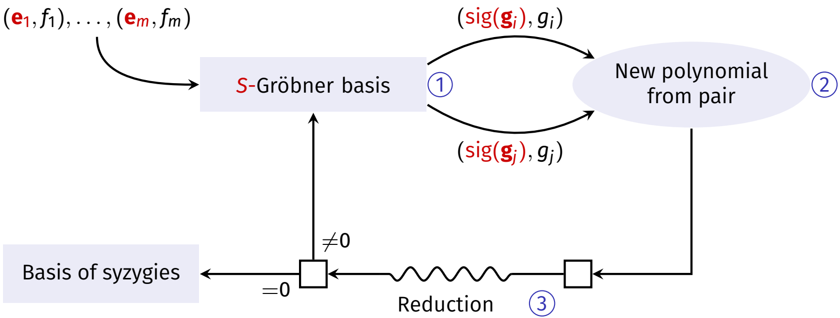
$\simeq \mu\mathbf{s}$ $\simeq \mu\mathbf{s}$ not regular

$$\begin{aligned}
\text{S-Pol}(\mathbf{f}, \mathbf{h}) &= \mu\mathbf{f} - \frac{a}{d}\mathbf{h} \\
&= \left(1 - \frac{ua}{d}\right)\mu\mathbf{f} - \frac{av}{d}\mu\mathbf{g} \\
&= \frac{vb}{d}\mu\mathbf{f} - \frac{av}{d}\nu\mathbf{g} \\
&= v\text{S-Pol}(\mathbf{f}, \mathbf{g})
\end{aligned}$$

$$\text{S-Pol}(\mathbf{g}, \mathbf{h}) = u\text{S-Pol}(\mathbf{f}, \mathbf{g})$$

Consequence: we need to allow all **non-singular** S-polynomials

Pan/Lichtblau's algorithm, with signatures (R is a PID)



1. **Selection**: non-decreasing signatures
2. **Construction**: non-singular S-polynomial if one of $\mathrm{lc}(\mathbf{g}_i)$ and $\mathrm{lc}(\mathbf{g}_j)$ divides the other

   or G-polynomial if $\mathrm{lc}(\mathbf{g}_i)$ and $\mathrm{lc}(\mathbf{g}_j)$ do not divide each other
3. **Reduction**: regular

Comparison of the algorithms

Theorem: criterion for correctness

Let $\mathcal{G} \subset \mathcal{I}$ and $\mathcal{G}_z \subset \mathrm{Syz}(I)$ be such that:

- for all i, there is an element with signature $\mathbf{e}_i$ in $\mathcal{G} \cup \mathcal{G}_z$ *"Correct ideal"*

Then $\mathcal{G}$ is a sig-Gröbner basis and $\mathcal{G}_z$ is a sig-basis of syzygies.

Kandri-Rody, Kapur	**Pan/Lichtblau**
S-pol if regular	S-pol if non-singular and lc divides
G-pol if lc does not divide	G-pol if lc does not divide
Regular reductions	Regular reductions

Comparison of the algorithms

Theorem: criterion for correctness

Let $\mathcal{G} \subset \mathcal{I}$ and $\mathcal{G}_z \subset \mathrm{Syz}(I)$ be such that:

- for all i, there is an element with signature $\mathbf{e}_i$ in $\mathcal{G} \cup \mathcal{G}_z$ *"Correct ideal"*
- all regular S-pols of $\mathcal{G}$ s-reduce to 0 mod $\mathcal{G}$ *"Gröbner basis"*

Then $\mathcal{G}$ is a sig-Gröbner basis and $\mathcal{G}_z$ is a sig-basis of syzygies.

Kandri-Rody, Kapur	**Pan/Lichtblau**
S-pol if regular	S-pol if non-singular and lc divides
G-pol if lc does not divide	G-pol if lc does not divide
Regular reductions	Regular reductions

Comparison of the algorithms

Theorem: criterion for correctness

Let $\mathcal{G} \subset \mathcal{I}$ and $\mathcal{G}_z \subset \mathrm{Syz}(I)$ be such that:

- for all i, there is an element with signature $\mathbf{e}_i$ in $\mathcal{G} \cup \mathcal{G}_z$ *"Correct ideal"*
- all regular S-pols of $\mathcal{G}$ s-reduce to 0 mod $\mathcal{G}$ *"Gröbner basis"*
- if those reductions are regular, their result is sig-reducible mod $\mathcal{G}_z$ *"Basis of syzygies"*

Then $\mathcal{G}$ is a sig-Gröbner basis and $\mathcal{G}_z$ is a sig-basis of syzygies.

Kandri-Rody, Kapur	**Pan/Lichtblau**
S-pol if regular	S-pol if non-singular and lc divides
G-pol if lc does not divide	G-pol if lc does not divide
Regular reductions	Regular reductions

Comparison of the algorithms

Theorem: criterion for correctness

Let $\mathcal{G} \subset \mathcal{I}$ and $\mathcal{G}_z \subset \mathrm{Syz}(I)$ be such that:

- for all i, there is an element with signature $\mathbf{e}_i$ in $\mathcal{G} \cup \mathcal{G}_z$ — *"Correct ideal"*
- all regular S-pols of $\mathcal{G}$ s-reduce to 0 mod $\mathcal{G}$ — *"Gröbner basis"*
- if those reductions are regular, their result is sig-reducible mod $\mathcal{G}_z$ — *"Basis of syzygies"*
- all G-pols of $\mathcal{G}$ are s-reducible mod $\mathcal{G}$
- all G-pols of $\mathcal{G}_z$ are sig-reducible mod $\mathcal{G}_z$

"Sufficiently many G-pols"

Then $\mathcal{G}$ is a sig-Gröbner basis and $\mathcal{G}_z$ is a sig-basis of syzygies.

Kandri-Rody, Kapur	**Pan/Lichtblau**
S-pol if regular	S-pol if non-singular and lc divides
G-pol if lc does not divide	G-pol if lc does not divide
Regular reductions	Regular reductions

Comparison of the algorithms

Theorem: criterion for correctness

Let $\mathcal{G} \subset \mathcal{I}$ and $\mathcal{G}_z \subset \mathrm{Syz}(I)$ be such that:

- for all i, there is an element with signature $\mathbf{e}_i$ in $\mathcal{G} \cup \mathcal{G}_z$ *"Correct ideal"*
- all regular S-pols of $\mathcal{G}$ s-reduce to 0 mod $\mathcal{G}$ *"Gröbner basis"*
- if those reductions are regular, their result is sig-reducible mod $\mathcal{G}_z$ *"Basis of syzygies"*
- all G-pols of $\mathcal{G}$ are s-reducible mod $\mathcal{G}$
- all G-pols of $\mathcal{G}_z$ are sig-reducible mod $\mathcal{G}_z$ *"Sufficiently many G-pols"*

Then $\mathcal{G}$ is a sig-Gröbner basis and $\mathcal{G}_z$ is a sig-basis of syzygies.

Kandri-Rody, Kapur	**Pan/Lichtblau**
S-pol if regular	S-pol if non-singular and lc divides
G-pol if lc does not divide	G-pol if lc does not divide
Regular reductions	Regular reductions
More criteria?	More criteria?

Super-reductibility

Super-reducible criterion in the case of fields

- $\mathbf{f}$ is super reducible modulo $\mathbf{g}$ if $t\mathrm{sig}(\mathbf{g}) \simeq \mathrm{sig}(\mathbf{f})$ and $t\mathrm{lt}(\mathbf{g}) = \mathrm{lt}(\mathbf{f})$
- $\mathbf{h} = \mathbf{f} - t\mathbf{g}$ is a singular s-reduction
- If $\mathbf{h}$ s-reduces to 0 mod $\mathcal{G}$, then $\mathbf{f}$ s-reduces to 0 mod $\mathcal{G}$
- **Consequence**: we can exclude super-reducible polynomials

Super-reducible criterion in the case of rings

- $\mathbf{f}$ is super reducible modulo $\mathbf{g}$ if $t\mathrm{sig}(\mathbf{g}) = \mathrm{sig}(\mathbf{f})$ and $t\mathrm{lt}(\mathbf{g}) \simeq \mathrm{lt}(\mathbf{f})$
- $\mathbf{f}' = \mathbf{f} - t\mathbf{g}$ is not a reduction!
- If $\mathbf{f}'$ s-reduces to 0 mod $\mathcal{G}$ and G-pols of $\mathcal{G}$ s-reduce to 0, then $\mathbf{f}$ s-reduces to 0 mod $\mathcal{G}$
- **Consequence**: we can exclude super-reducible S-polynomials

Cover property

Definition: cover property in the case of fields

The pair $(\mathbf{f}_1, \mathbf{f}_2)$ is covered by $\mathbf{g} \in \mathcal{G} \cup \mathcal{G}_z$ if:

- there exists a term t such that $\mathrm{sig}(\text{S-Pol}(\mathbf{f}_1, \mathbf{f}_2)) = t\,\mathrm{sig}(\mathbf{g})$
- $t\,\mathrm{lt}(\mathbf{g}) < \mathrm{lcmlm}(\mathbf{f}_1, \mathbf{f}_2)$ (with $\mathrm{lt}(\mathbf{g}) = 0$ if syzygy)

Cover property

Definition: cover property in the case of fields

The pair $(\mathbf{f}_1, \mathbf{f}_2)$ is covered by $\mathbf{g} \in \mathcal{G} \cup \mathcal{G}_z$ if:

- there exists a term t such that $\mathrm{sig}(\text{S-Pol}(\mathbf{f}_1, \mathbf{f}_2)) = t\mathrm{sig}(\mathbf{g})$
- $t\mathrm{lt}(\mathbf{g}) < \mathrm{lcmlm}(\mathbf{f}_1, \mathbf{f}_2)$ (with $\mathrm{lt}(\mathbf{g}) = 0$ if syzygy)

Definition: cover property in the case of rings

The pair $(\mathbf{f}_1, \mathbf{f}_2)$ is covered by $\mathbf{g} \in \mathcal{G}$ and $\mathbf{z} \in \mathcal{G}_z$ if:

- there exist terms t_g, t_z such that $\mathrm{sig}(\text{S-Pol}(\mathbf{f}_1, \mathbf{f}_2)) = t_g\mathrm{sig}(\mathbf{g}) + t_z\mathrm{sig}(\mathbf{z})$
- $t_g\mathrm{lt}(\mathbf{g}) < \mathrm{lcmlm}(\mathbf{f}_1, \mathbf{f}_2)$

Correctness criterion with the cover property

Reminder: general criterion for correctness

Let $\mathcal{G} \subset \mathcal{I}$ and $\mathcal{G}_z \subset \mathrm{Syz}(I)$ be such that:

- for all i, there is an element with signature $\mathbf{e}_i$ in $\mathcal{G} \cup \mathcal{G}_z$
- all regular S-pols of $\mathcal{G}$ s-reduce to 0 mod $\mathcal{G}$
- if those reductions are regular, their result is sig-reducible mod $\mathcal{G}_z$
- all G-pols of $\mathcal{G}$ are s-reducible mod $\mathcal{G}$
- all G-pols of $\mathcal{G}_z$ are sig-reducible mod $\mathcal{G}_z$

Then $\mathcal{G}$ is a sig-Gröbner basis and $\mathcal{G}_z$ is a sig-basis of syzygies.

Correctness criterion with the cover property

Theorem: cover criterion for correctness

Let $\mathcal{G} \subset \mathcal{I}$ and $\mathcal{G}_z \subset \mathrm{Syz}(I)$ be such that:

- for all i, there is an element with signature $\mathbf{e}_i$ in $\mathcal{G} \cup \mathcal{G}_z$
- all regular S-pols of $\mathcal{G}$ are covered by a pair of $\mathcal{G}$, $\mathcal{G}_z$
- all G-pols of $\mathcal{G}$ are s-reducible modulo $\mathcal{G}$
- all G-pols of $\mathcal{G}_z$ are sig-reducible mod $\mathcal{G}_z$

Then $\mathcal{G}$ is a SGB and $\mathcal{G}_z$ is a sig-basis of syzygies.

This criterion is convenient...

- in practice, because it allows to eliminate many elements
- in theory, because it allows for a simpler proof of correctness

But it requires that all regular S-pols of $\mathcal{G}$ be covered, which Pan/Lichtblau a priori cannot enforce.

Quantitative comparison between the algorithms

System	Algorithm	Total pairs	Reduced	To zero	Time (s)
Katsura-4	Kandri-Rody, Kapur	420	188	0	1.35
	Pan/Lichtblau	855	412	0	1.6
Katsura-5	Kandri-Rody, Kapur	248	723	0	32.40
	Pan/Lichtblau	7178	3983	0	79.87
Cyclic-5	Kandri-Rody, Kapur	221	63	0	0.37
	Pan/Lichtblau	347	158	0	0.71
Cyclic-6	Kandri-Rody, Kapur	3019	742	8	200.33
	Pan/Lichtblau	9672	5782	8	616.82

- Toy implementation of both algorithms in Magma
- Available at `https://gitlab.com/thibaut.verron/signature-groebner-rings`
- Kandri-Rody and Kapur is almost always more efficient than Pan/Lichtblau
- It is not due to the lack of cover criterion

Indicative timings

Operations

- Gröbner basis: signatures (Kandri-Rody and Kapur) vs Magma's `GroebnerBasis` (F4)
- GB with coefs.: signature reconstruction vs Magma's `IdealWithFixedBasis` (F4 + tracking)
- Basis of syzygy module: signature reconstruction vs Magma's `SyzygyMatrix` (module GB)

System	S-GB (s)	Recons. (s)	**Total (s)**	GB (s)	**GB + coefs (s)**	**Syz. basis (s)**
Cyclic-5	0.4	0.1	**0.5**	0.01	**954.6**	**954.8**
Cyclic-6	200.3	10.6	**210.9**	2.08	**>24h**	**>24h**

Conclusion of part 1

This work

- Two signature-based algorithms for PID's following closely Buchberger's algorithm
- Compatible with powerful criteria such as super-reducibility and the cover criterion
- Additional criteria and optimizations are available (coprime criterion, Gebauer-Möller criteria, coefficient reductions...)
- Toy implementation in Magma

Future directions

- Linear algebra algorithms à la F4
- Improve implementation
- Extend use of signature bases

More details and references

- Francis and Verron, *On Two Signature Variants Of Buchberger's Algorithm Over Principal Ideal Domains*, ISSAC 2021

Part 2: signature Gröbner bases in the free algebra

Joint work with Clemens Hofstadler

Non-commutative Gröbner bases

Context:

- R field
- $A = R\langle X_1, \ldots, X_n \rangle$ free algebra over R
- Monomials are words $X_{i_1} X_{i_2} \cdots X_{i_d}$

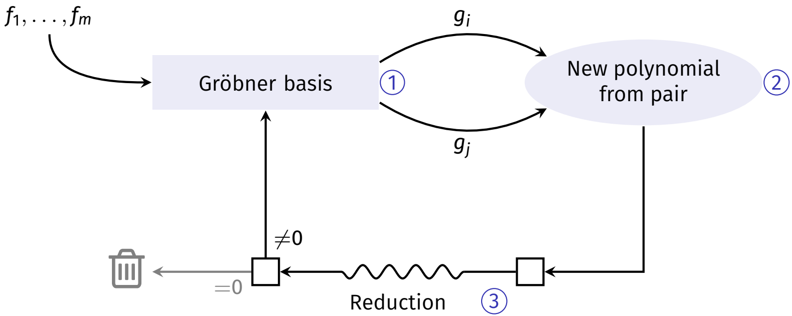
Gröbner bases:

- Monomial ordering and reduction are defined as usual
- Gröbner bases are defined as usual

Particularity:

- The free algebra is not Noetherian
- Most ideals do not admit a finite Gröbner basis
- It is not decidable whether an ideal admits a finite Gröbner basis

Non-commutative Buchberger's algorithm



1. **Selection**: fair selection strategy
2. **Construction**: S-polynomials
3. **Reduction**

Constructions in the non-commutative case

Several ways to make S-polynomials

- **Overlap ambiguity**

$$f = {\color{green}\rule{2em}{0.6em}}{\color{red}\rule{2em}{0.6em}}\hphantom{\rule{2em}{0.6em}} + \cdots \qquad \mathrm{SPol}(f,g) = f\,{\color{cyan}\rule{2em}{0.6em}} - {\color{green}\rule{2em}{0.6em}}\,g$$
$$g = \hphantom{\rule{2em}{0.6em}}{\color{red}\rule{2em}{0.6em}}{\color{cyan}\rule{2em}{0.6em}} + \cdots$$

- **Inclusion ambiguity**

$$f = \hphantom{\rule{2em}{0.6em}}{\color{red}\rule{2em}{0.6em}}\hphantom{\rule{2em}{0.6em}} + \cdots \qquad \mathrm{SPol}(f,g) = {\color{green}\rule{2em}{0.6em}}\,f\,{\color{cyan}\rule{2em}{0.6em}} - g$$
$$g = {\color{green}\rule{2em}{0.6em}}{\color{red}\rule{2em}{0.6em}}{\color{cyan}\rule{2em}{0.6em}} + \cdots$$

Constructions in the non-commutative case

Several ways to make S-polynomials

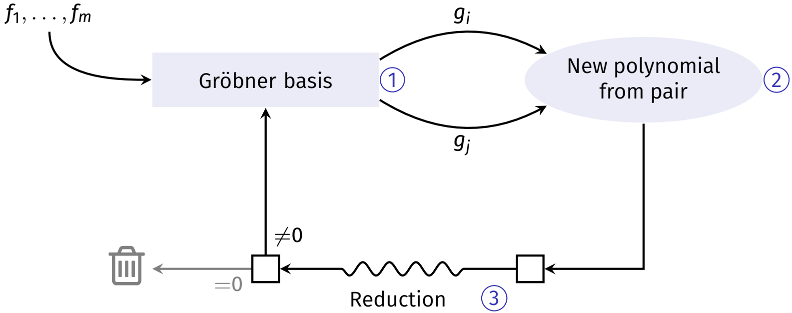
- **Overlap ambiguity**

- **Inclusion ambiguity**

Remarks:

- The combination need not be minimal, and S-polynomials are not unique!
- *xyxy* has an (overlap) ambiguity with itself: *xyxy* / *xyxy*
- *xxyx* and *xy* have two ambiguities: *xxyx* / *xy*   *xxyx* / *xy*
- Two polynomials can only give rise to finitely many S-polynomials
- It is required that the central part is non-trivial (coprime criterion)

Non-commutative Buchberger's algorithm



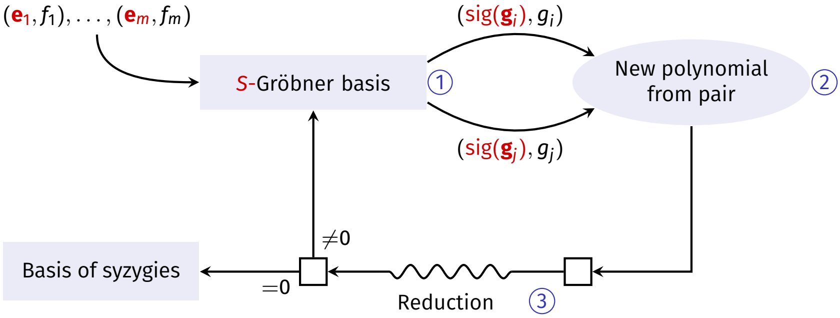
1. **Selection**: fair selection strategy *"Every S-polynomial is selected eventually."*
2. **Construction**: S-polynomials
3. **Reduction**

Signatures for non-commutative polynomials

Setting:

- Bimodule $M = A\mathbf{e}_1 A \oplus \cdots \oplus A\mathbf{e}_m A$ with the usual morphism $M \to A$ with image I
- Equipped with a module monomial ordering
- We require the ordering to be fair (isomorphic to $\mathbb{N}$)
- Signature of $\mathbf{f}$ = leading monomial of the module part of $\mathbf{f}$
- Regular and singular operations are defined as before

Non-commutative Buchberger's algorithm with signatures



1. **Selection**: non-decreasing signatures for a fair ordering
2. **Construction**: regular S-polynomials
3. **Reduction** (regular)

Termination

Question 1: does the algorithm terminate?

- Of course not, because some ideals do not have a finite Gröbner basis.

Termination

Question 1: does the algorithm terminate?

- Of course not, because some ideals do not have a finite Gröbner basis.

Question 2: okay, but what if they do?

- Still not. In most cases, the module of syzygies does not have a finite Gröbner basis
- Conjecture: it's always the case if $n > 1$ (non-commutative) and $m > 1$ (non-principal)

Termination: trivial syzygies

Question 1: does the algorithm terminate?

- Of course not, because some ideals do not have a finite Gröbner basis.

Question 2: okay, but what if they do?

- Still not. In most cases, the module of syzygies does not have a finite Gröbner basis
- Conjecture: it's always the case if $n > 1$ (non-commutative) and $m > 1$ (non-principal)

Obstruction: trivial syzygies!

- Syzygies of the form $\mathbf{f} \blacksquare g - f \blacksquare \mathbf{g}$ for any monomial $\blacksquare$
- Signature: $\max\big(\mathrm{sig}(\mathbf{f}) \blacksquare \mathrm{lm}(g), \mathrm{lm}(f) \blacksquare \mathrm{sig}(\mathbf{g})\big)$
- Because $\blacksquare$ is put in the middle, there is no reason to expect this set to be finitely generated!

Termination: trivial syzygies and how to find them

Question 1: does the algorithm terminate?

- Of course not, because some ideals do not have a finite Gröbner basis.

Question 2: okay, but what if they do?

- Still not. In most cases, the module of syzygies does not have a finite Gröbner basis
- Conjecture: it's always the case if $n > 1$ (non-commutative) and $m > 1$ (non-principal)

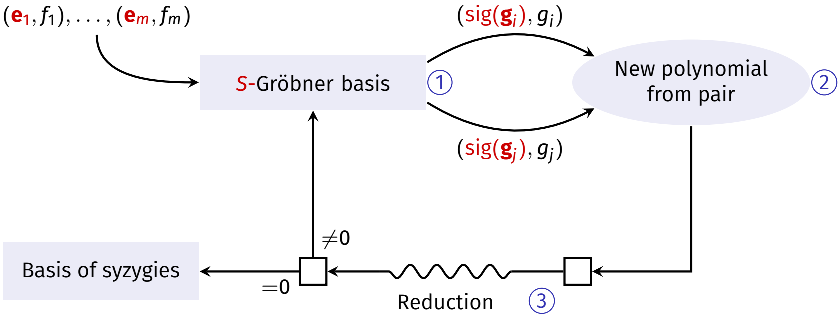
Obstruction: trivial syzygies!

- Syzygies of the form $\mathbf{f} \blacksquare g - f \blacksquare \mathbf{g}$ for any monomial $\blacksquare$
- Signature: $\max\big(\mathrm{sig}(\mathbf{f}) \blacksquare \mathrm{lm}(g), \mathrm{lm}(f) \blacksquare \mathrm{sig}(\mathbf{g})\big)$
- Because $\blacksquare$ is put in the middle, there is no reason to expect this set to be finitely generated!

Solution: signatures!

- Identifying trivial syzygies is what signatures were made for! (F5 criterion)
- In the commutative case, this is an optimization
- In the non-commutative case, it is a requirement

Non-commutative Buchberger's algorithm with signatures



1. **Selection**: non-decreasing signatures for a fair ordering
2. **Construction**: regular S-polynomials which are not eliminated by the F5 criterion
3. **Reduction** (regular)

What do we get?

Output of the algorithm: a signature Gröbner basis, allowing to recover

- a sig-Gröbner basis $\mathcal{G}$ (with coordinates)
- a set $\mathcal{H}$ of syzygies such that $\mathcal{H} \cup \{\text{trivial syzygies of } \mathcal{G}\}$ is a basis of the module of syzygies
- a way to test if any module monomial is the leading term of a syzygy (trivial or not)

Results:

- The algorithm enumerates such a signature Gröbner basis
- The algorithm terminates iff the ideal admits a finite signature Gröbner basis
- This implies that the ideal admits a finite GB and a finite “basis of non-trivial syzygies” $\mathcal{H}$
- **Conjecture**: the converse holds

This is the first algorithm producing an effective representation of some modules of syzygies in the free algebra!

Implementation

What we have

- Toy implementation in Mathematica
- Part of the package `OperatorGB`, available at `https://clemenshofstadler.com/software/`
- Too slow to report on timings

Particularity

- The F5 criterion is necessary to maximize the chances of the algorithm terminating
- The PoT ordering is not fair
- The F5 criterion is expensive! (quadratic in the size of $\mathcal{G}$)

Conclusion of part 2

This work

- Signature-based algorithm for enumerating signature Gröbner bases in the free algebra
- Terminates whenever a finite signature Gröbner basis exists
- Unlike the commutative case, taking care of trivial syzygies is more than an optimization
- Effective and finite representation of the module of syzygies in some non-trivial cases

Open questions and future directions

- Improve implementation
- Conjecture on characterization of existence of finite signature Gröbner basis
- Free algebra over $\mathbb{Z}$? (worse than the worst of both worlds)
- Application to the computation of short representations
- Computations in quotients of the algebra

More details and references

- Hofstadler and Verron, *Signature Gröbner bases, bases of syzygies and cofactor reconstruction in the free algebra*, `ArXiV:2107.14675`