Exploiting algebraic structures to solve polynomial systems of equations

Brice Boyer, Christian Eder, Jean-Charles Faugère, Fayssal Martani, John Perry and Bjarke Hammersholt Roune

Séminaire de théorie des codes et cryptographie à Neuchâtel (et à Zürich)

May 18, 2015
Conventions

- $\mathcal{R} = \mathcal{K}[x_1, \ldots, x_n]$, $\mathcal{K}$ field, $<$ well-ordering on $\text{Mon}(x_1, \ldots, x_n)$

- $f \in \mathcal{R}$ can be represented in a unique way by $<$. $\Rightarrow$ Definitions as $\text{lc}(f)$, $\text{lm}(f)$, and $\text{lt}(f)$ make sense.

- An ideal $I$ in $\mathcal{R}$ is an additive subgroup of $\mathcal{R}$ such that for $f \in I$, $g \in \mathcal{R}$ it holds that $fg \in I$.

- $G = \{g_1, \ldots, g_s\} \subset \mathcal{R}$ is a Gröbner basis for $I = \langle f_1, \ldots, f_m \rangle$ w.r.t. $<$

$$
\iff
G \subset I \text{ and } L_<(G) = L_<(I)
$$
1. A lot of crypto systems boil down to find a solution (a finite number of solutions) of a system of polynomial equations.
1. A lot of crypto systems boil down to find a solution (a finite number of solutions) of a system of polynomial equations.

2. For example, multivariate crypto systems like (Multi-)HFE(+-), UOV or Rainbow

3. Minrank \((n, k, r)\) problem: Given matrices \(M_0, \ldots, M_k \in \mathbb{M}_{n \times n}(\mathbb{K})\), find (if possible) \((\lambda_1, \ldots, \lambda_k) \in \mathbb{K}^k\) such that

\[
\text{rank} \left( \sum_{i=1}^{k} \lambda_i M_i - M_0 \right) \leq r.
\]
Why?

1. A lot of crypto systems boil down to find a solution (a finite number of solutions) of a system of polynomial equations.

2. For example, multivariate crypto systems like (Multi-)HFE(+-), UOV or Rainbow

3. Minrank \((n, k, r)\) problem: Given matrices \(M_0, \ldots, M_k \in M_{n \times n}(k)\), find (if possible) \((\lambda_1, \ldots, \lambda_k) \in k^n\) such that

\[
\text{rank} \left( \sum_{i=1}^{k} \lambda_i M_i - M_0 \right) \leq r.
\]

Solving polynomial equations is important
Gröbner Bases are cool!
Buchberger’s criterion

**S-polynomials**
Let $f \neq 0, g \neq 0 \in \mathbb{R}$ and let $\lambda = \text{lcm}(\text{lt}(f), \text{lt}(g))$ be the least common multiple of $\text{lt}(f)$ and $\text{lt}(g)$. The **S-polynomial** between $f$ and $g$ is given by

$$\text{spol}(f, g) := \frac{\lambda}{\text{lt}(f)} f - \frac{\lambda}{\text{lt}(g)} g.$$
Buchberger’s criterion

**S-polynomials**
Let \( f \neq 0, g \neq 0 \in \mathbb{R} \) and let \( \lambda = \text{lcm} (\text{lt}(f), \text{lt}(g)) \) be the least common multiple of \( \text{lt}(f) \) and \( \text{lt}(g) \). The **S-polynomial** between \( f \) and \( g \) is given by

\[
\text{spol}(f, g) := \frac{\lambda}{\text{lt}(f)} f - \frac{\lambda}{\text{lt}(g)} g.
\]

**Buchberger’s criterion** [1]
Let \( I = \langle f_1, \ldots, f_m \rangle \) be an ideal in \( \mathbb{R} \). A finite subset \( G \subset \mathbb{R} \) is a **Gröbner basis** for \( I \) if \( G \subset I \) and for all \( f, g \in G \) : \( \text{spol}(f, g) \xrightarrow{G} 0. \)
Buchberger’s algorithm

Input: Ideal \( I = \langle f_1, \ldots, f_m \rangle \)
Output: Gröbner basis \( G \) for \( I \)

1. \( G \leftarrow \emptyset \)
2. \( G \leftarrow G \cup \{ f_i \} \) for all \( i \in \{1, \ldots, m \} \)
3. Set \( P \leftarrow \{ \text{spol} (f_i, f_j) \mid f_i, f_j \in G, i > j \} \)
Buchberger’s algorithm

**Input:** Ideal \( I = \langle f_1, \ldots, f_m \rangle \)

**Output:** Gröbner basis \( G \) for \( I \)

1. \( G \leftarrow \emptyset \)
2. \( G \leftarrow G \cup \{ f_i \} \) for all \( i \in \{1, \ldots, m \} \)
3. Set \( P \leftarrow \{ \text{spol} (f_i, f_j) \mid f_i, f_j \in G, i > j \} \)
4. Choose \( p \in P, P \leftarrow P \setminus \{ p \} \)
Buchberger’s algorithm

**Input:** Ideal $I = \langle f_1, \ldots, f_m \rangle$

**Output:** Gröbner basis $G$ for $I$

1. $G \leftarrow \emptyset$

2. $G \leftarrow G \cup \{f_i\}$ for all $i \in \{1, \ldots, m\}$

3. Set $P \leftarrow \{\text{spol}(f_i, f_j) \mid f_i, f_j \in G, i > j\}$

4. Choose $p \in P$, $P \leftarrow P \setminus \{p\}$
   
   (a) If $p \xrightarrow{G} 0$ ➤ **no new information**
   
   Go on with the next element in $P$.

   (b) If $p \xrightarrow{G} q \neq 0$ ➤ **new information**
   
   Build new S-pair with $q$ and add them to $P$.
   
   Add $q$ to $G$.
   
   Go on with the next element in $P$.

5. When $P = \emptyset$ we are done and $G$ is a Gröbner basis for $I$. 
How to improve computations?

- Modular computations $\mathbb{Q} \rightarrow$ several $\mathbb{Z}_{p_i}$ computations and CRT
- Predict zero reductions fast checks $\rightarrow$ fewer useless reductions
- Sort pair set selection of pairs, degree drops, mutants, etc.
- Homogenization $d$-Gröbner bases, sugar degree
- Change of order transformation to different monomial order
- Linear Algebra (specialized) Gaussian Elimination
- Sparse Gröbner Bases exploitation of sparsity, Newton polygons
- ...
How to improve computations?

- **Predict zero reductions** fast checks $\rightarrow$ fewer useless reductions

- **Linear Algebra** (specialized) Gaussian Elimination
● Predicting zero reductions

● Fast linear algebra for computing Gröbner bases
How to detect zero reductions in advance?

Let $\langle g_1, g_2 \rangle \in \mathbb{Q}[x, y, z]$ and let $<$ denote the reverse lexicographical ordering. Let

$$g_1 = xy - z^2, \quad g_2 = y^2 - z^2$$
How to detect zero reductions in advance?

Let \( I = \langle g_1, g_2 \rangle \in \mathbb{Q}[x, y, z] \) and let \( < \) denote the reverse lexicographical ordering. Let

\[
\begin{align*}
g_1 &= xy - z^2, \\
g_2 &= y^2 - z^2
\end{align*}
\]

\[
\text{spol}(g_2, g_1) = xg_2 - yg_1 = xy^2 - xz^2 - xy^2 + yz^2 = -xz^2 + yz^2.
\]

\[
\implies g_3 = xz^2 - yz^2.
\]
How to detect zero reductions in advance?

Let \( l = \langle g_1, g_2 \rangle \in \mathbb{Q}[x, y, z] \) and let \(<\) denote the reverse lexicographical ordering. Let

\[
g_1 = xy - z^2, \quad g_2 = y^2 - z^2
\]

\[
\text{spol}(g_2, g_1) = xg_2 - yg_1 = xy^2 - xz^2 - xy^2 + yz^2 = -xz^2 + yz^2.
\]

\[
\implies g_3 = xz^2 - yz^2.
\]

\[
\text{spol}(g_3, g_1) = xyz^2 - y^2z^2 - xyz^2 + z^4 = -y^2z^2 + z^4.
\]
How to detect zero reductions in advance?

Let \( I = \langle g_1, g_2 \rangle \in \mathbb{Q}[x, y, z] \) and let \(<\) denote the reverse lexicographical ordering. Let

\[
g_1 = xy - z^2, \quad g_2 = y^2 - z^2
\]

Then

\[
\text{spol}(g_2, g_1) = xg_2 - yg_1 = xy^2 - xz^2 - xy^2 + yz^2 = -xz^2 + yz^2.
\]

\[\longrightarrow g_3 = xz^2 - yz^2.\]

Then

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

We can reduce further using \( z^2 g_2 \):

\[-y^2 z^2 + z^4 + y^2 z^2 - z^4 = 0.\]
How to detect zero reductions in advance?

Can we see something? How are the generators of the S-polynomials related to each other?
How to detect zero reductions in advance?

Can we see something? How are the generators of the S-polynomials related to each other?

\[
\text{spol}(g_3, g_2) = y^2 (xz^2 - yz^2) - xz^2 (y^2 - z^2)
\]
\[
= \text{lt}(g_2)g_3 - \text{lt}(g_3)g_2
\]
\[
= \text{lt}(g_2)\text{lot}(g_3) - \text{lt}(g_3)\text{lot}(g_2)
\]
How to detect zero reductions in advance?

Can we see something? How are the generators of the S-polynomials related to each other?

\[ \text{spol}(g_3, g_2) = y^2 (xz^2 - yz^2) - xz^2 (y^2 - z^2) \]

\[ = \text{lt}(g_2) g_3 - \text{lt}(g_3) g_2 \]

\[ = \text{lt}(g_2) \text{lot}(g_3) - \text{lt}(g_3) \text{lot}(g_2) \]

For all \( u \in \text{support}(\text{lot}(g_3)) \) we can reduce with \( u g_2 \):

\[ \Rightarrow \text{lt}(g_2) \text{lot}(g_3) - g_2 \text{lot}(g_3) - \text{lt}(g_3) \text{lot}(g_2) \]

\[ = - \text{lot}(g_2) \text{lot}(g_3) - \text{lt}(g_3) \text{lot}(g_2) \]

\[ = - g_3 \text{lot}(g_2). \]
How to detect zero reductions in advance?

Can we see something? How are the generators of the S-polynomials related to each other?

$$\text{spol}(g_3, g_2) = y^2 (xz^2 - yz^2) - xz^2 (y^2 - z^2)$$

$$= \text{lt}(g_2)g_3 - \text{lt}(g_3)g_2$$

$$= \text{lt}(g_2)\text{lot}(g_3) - \text{lt}(g_3)\text{lot}(g_2)$$

For all $$u \in \text{support}(\text{lot}(g_3))$$ we can reduce with $$ug_2$$:

$$\Rightarrow \text{lt}(g_2)\text{lot}(g_3) - g_2\text{lot}(g_3) - \text{lt}(g_3)\text{lot}(g_2)$$

$$= - \text{lot}(g_2)\text{lot}(g_3) - \text{lt}(g_3)\text{lot}(g_2)$$

$$= - g_3\text{lot}(g_2).$$

So we can reduce this to zero by $$vg_3$$ for all $$v \in \text{support}(\text{lot}(g_2)).$$
Buchberger’s criteria

**Product criterion [2]**

If $\text{lcm}(\text{lt}(f), \text{lt}(g)) = \text{lt}(f) \text{lt}(g)$ then $\text{spol}(f, g) \xrightarrow{\{f, g\}} 0$. 
Buchberger’s criteria

**Product criterion [2]**

If \( \text{lcm} (\text{lt}(f), \text{lt}(g)) = \text{lt}(f) \text{lt}(g) \) then \( \text{spol}(f, g) \xrightarrow{\{f,g\}} 0. \)

Couldn’t we remove \( \text{spol}(g_3, g_2) \) in a different way?
Buchberger’s criteria

**Product criterion [2]**

If \( \text{lcm}(\text{lt}(f), \text{lt}(g)) = \text{lt}(f) \text{lt}(g) \) then \( \text{spol}(f, g) \rightarrow 0 \).

Couldn’t we remove \( \text{spol}(g_3, g_2) \) in a different way?

\[
\text{lt}(g_1) = xy \mid xy^2 z^2 = \text{lcm}(\text{lt}(g_3), \text{lt}(g_2))
\]
Buchberger’s criteria

**Product criterion [2]**

If \( \text{lcm}(\text{lt}(f), \text{lt}(g)) = \text{lt}(f) \text{lt}(g) \) then \( \text{spol}(f, g) \xrightarrow{\{f,g\}} 0. \)

Couldn’t we remove \( \text{spol}(g_3, g_2) \) in a different way?

\[
\text{lt}(g_1) = xy \mid xy^2 z^2 = \text{lcm}(\text{lt}(g_3), \text{lt}(g_2))
\]

\[
\implies \text{We can rewrite } \text{spol}(g_3, g_2):
\]

\[
\text{spol}(g_3, g_2) = y \text{ spol}(g_3, g_1) - z^2 \text{ spol}(g_2, g_1) = y(yg_3 - z^2 g_1) - z^2 (xg_2 - yg_1)
\]

\[
\xrightarrow{G \leftarrow 0} \quad \xrightarrow{G \leftarrow -g_3}
\]
Buchberger’s criteria

**Product criterion [2]**

If $\text{lcm}(\text{lt}(f), \text{lt}(g)) = \text{lt}(f) \text{lt}(g)$ then $\text{spol}(f, g) \xrightarrow{\{f, g\}} 0$.

Could we remove $\text{spol}(g_3, g_2)$ in a different way?

$$\text{lt}(g_1) = xy \mid xy^2 z^2 = \text{lcm}(\text{lt}(g_3), \text{lt}(g_2))$$

$$\implies \text{We can rewrite } \text{spol}(g_3, g_2):$$

$$\text{spol}(g_3, g_2) = y \text{spol}(g_3, g_1) - z^2 \text{spol}(g_2, g_1) = y(yg_3 - z^2 g_1) - z^2(xg_2 - yg_1)$$

Once we have reduced $\text{spol}(g_2, g_1)$ and $\text{spol}(g_3, g_1)$ we do not need to reduce $\text{spol}(g_3, g_2)$. 
Buchberger’s criteria

**Chain criterion [3]**
Let \( f, g, h \in R, G \subset R \) finite. If

1. \( \text{lt}(h) \mid \text{lcm}(\text{lt}(f), \text{lt}(g)) \), and

2. \( \text{spol}(f, h) \) and \( \text{spol}(h, g) \) have a standard representation w.r.t. \( G \) respectively,

then \( \text{spol}(f, g) \) has a standard representation w.r.t. \( G \).
Buchberger’s criteria

Chain criterion [3]
Let \( f, g, h \in R \), \( G \subset R \) finite. If

1. \( \text{lcm}(\text{lt}(f), \text{lt}(g)) \mid \text{lt}(h) \), and
2. \( \text{spol}(f, h) \) and \( \text{spol}(h, g) \) have a standard representation w.r.t. \( G \) respectively,

then \( \text{spol}(f, g) \) has a standard representation w.r.t. \( G \).

Note
Do not remove too much information! If \( \lambda = 1 \) and

\[
\text{spol}(f, g) = \lambda \text{spol}(f, h) + \sigma \text{spol}(h, g),
\]

then we can remove \( \text{spol}(f, g) \) or \( \text{spol}(f, h) \) but not both!
Buchberger’s criteria

**Chain criterion [3]**
Let $f, g, h \in \mathbb{R}, G \subset \mathbb{R}$ finite. If

1. $\text{lt}(h) \mid \text{lcm}(\text{lt}(f), \text{lt}(g))$, and

2. $\text{spol}(f, h)$ and $\text{spol}(h, g)$ have a standard representation w.r.t. $G$ respectively,
then $\text{spol}(f, g)$ has a standard representation w.r.t. $G$.

**Note**
Do not remove too much information! If $\lambda = 1$ and

$$\text{spol}(f, g) = \lambda \text{spol}(f, h) + \sigma \text{spol}(h, g),$$

then we can remove $\text{spol}(f, g)$ or $\text{spol}(f, h)$ but not both!

Combine both criteria using Gebauer-Möller’s installation [8].
Can we do even better?

In our example we still need to consider

$$\text{spol}(g_3, g_1) \xrightarrow{G} 0.$$ 

How to get rid of this useless computation?
Can we do even better?

In our example we still need to consider

$$\text{spol}(g_3, g_1) \xrightarrow{G} 0.$$ 

How to get rid of this useless computation?

Use more structure of $I \rightarrow \text{Signatures}$
Let \( I = \langle f_1, \ldots, f_m \rangle \subset R \).

**Idea:** Give each \( f \in I \) a bit more structure:
Let $I = \langle f_1, \ldots, f_m \rangle \subset \mathcal{R}$.

**Idea:** Give each $f \in I$ a bit more structure:

1. Let $\mathcal{R}^m$ be generated by $e_1, \ldots, e_m$ and let $\prec$ be a compatible monomial order on the monomials of $\mathcal{R}^m$. 
Let $I = \langle f_1, \ldots, f_m \rangle \subset \mathcal{R}$.

**Idea:** Give each $f \in I$ a bit more structure:

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

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

Let \( l = \langle f_1, \ldots, f_m \rangle \subset \mathcal{R} \).

**Idea:** Give each \( f \in l \) a bit more structure:

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

2. Let \( \alpha \mapsto \overline{\alpha} : \mathcal{R}^m \rightarrow \mathcal{R} \) such that \( e_i = f_i \) for all \( i \).

3. Each \( f \in l \) can be represented via some \( \alpha \in \mathcal{R}^m : f = \overline{\alpha} \).
Let $l = \langle f_1, \ldots, f_m \rangle \subset \mathcal{R}$.

**Idea:** Give each $f \in l$ a bit more structure:

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

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

3. Each $f \in l$ can be represented via some $\alpha \in \mathcal{R}^m$: $f = \overline{\alpha}$

4. A **signature** of $f$ is given by $s(f) = \text{lt}_{\prec}(\alpha)$ where $f = \overline{\alpha}$.
Signatures

Let \( I = \langle f_1, \ldots, f_m \rangle \subset R \).

**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 \( \alpha \mapsto \overrightarrow{\alpha} : R^m \to R \) such that \( e_i = f_i \) for all \( i \).

3. Each \( f \in I \) can be represented via some \( \alpha \in R^m \): \( f = \alpha \).

4. A **signature** of \( f \) is given by \( s(f) = \text{lt}_\prec(\alpha) \) where \( f = \overrightarrow{\alpha} \).

5. An element \( \alpha \in R^m \) such that \( \overrightarrow{\alpha} = 0 \) is called a **syzygy**.
Our example again – with signatures and \( \rightsimeq_{\text{pot}} \)

\[
\begin{align*}
  g_1 &= xy - z^2, \quad s(g_1) = e_1, \\
  g_2 &= y^2 - z^2, \quad s(g_2) = e_2.
\end{align*}
\]
Our example again – with signatures and $\prec_{\text{pot}}$

\[ g_1 = xy - z^2, \quad s(g_1) = e_1, \]
\[ g_2 = y^2 - z^2, \quad s(g_2) = e_2. \]

\[ g_3 = \text{spol}(g_2, g_1) = xg_2 - yg_1 \]
\[ \Rightarrow s(g_3) = x s(g_2) = xe_2. \]
Our example again – with signatures and $\prec_{\text{pot}}$

\[ g_1 = xy - z^2, \ s(g_1) = e_1, \]
\[ g_2 = y^2 - z^2, \ s(g_2) = e_2. \]

\[ g_3 = \text{spol}(g_2, g_1) = xg_2 - yg_1 \]
\[ \Rightarrow s(g_3) = xs(g_2) = xe_2. \]

\[ \text{spol}(g_3, g_1) = yg_3 - z^2 g_1 \]
\[ \Rightarrow s(\text{spol}(g_3, g_1)) = ys(g_3) = xye_2. \]
Our example again – with signatures and $\prec_{pot}$

$g_1 = xy - z^2$, $s(g_1) = e_1$,

$g_2 = y^2 - z^2$, $s(g_2) = e_2$.

$g_3 = \text{spol}(g_2, g_1) = xg_2 - yg_1$

$\Rightarrow s(g_3) = xs(g_2) = xe_2$.

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

$\Rightarrow s(\text{spol}(g_3, g_1)) = ys(g_3) = xye_2$.

Note that $s(\text{spol}(g_3, g_1)) = xye_2$ and $\text{lm}(g_1) = xy$. 
Think in the module

\[ \alpha \in \mathbb{R}^m \rightarrow \text{polynomial } \overline{\alpha} \text{ with } \text{lt} (\overline{\alpha}), \text{ signature } s(\alpha) = \text{lt} (\alpha) \]
Think in the module

\[ \alpha \in \mathbb{R}^m \longrightarrow \text{polynomial } \overline{\alpha} \text{ with } \text{lt}(\overline{\alpha}), \text{signature } \sigma(\alpha) = \text{lt}(\alpha) \]

S-pairs/S-polynomials:

\[ \text{spol}(\overline{\alpha}, \overline{\beta}) = a\overline{\alpha} - b\overline{\beta} \quad \Rightarrow \quad \text{spair}(\alpha, \beta) = a\alpha - b\beta \]
Think in the module

\( \alpha \in \mathbb{R}^m \rightarrow \) polynomial \( \overline{\alpha} \) with \( \text{lt} (\overline{\alpha}) \), signature \( s(\alpha) = \text{lt}(\alpha) \)

S-pairs/S-polynomials:

\[
\text{spol} (\overline{\alpha}, \overline{\beta}) = a\overline{\alpha} - b\overline{\beta} \quad \rightarrow \quad \text{spair} (\alpha, \beta) = a\alpha - b\beta
\]

\( s \)-reductions:

\[
\overline{\gamma} - d\overline{\delta} \quad \rightarrow \quad \gamma - d\delta
\]
Think in the module

\[ \alpha \in \mathbb{R}^m \implies \text{polynomial } \overline{\alpha} \text{ with } \text{lt}(\overline{\alpha}), \text{signature } s(\alpha) = \text{lt}(\alpha) \]

S-pairs/S-polynomials:

\[ \text{spol}(\overline{\alpha}, \overline{\beta}) = a\overline{\alpha} - b\overline{\beta} \implies \text{spair}(\alpha, \beta) = a\alpha - b\beta \]

\(s\)-reductions:

\[ \overline{\gamma} - d\overline{\delta} \implies \gamma - d\delta \]

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

We pick from \( P \) by increasing signature.
Signature-based criteria

\[ s(\alpha) = s(\beta) \implies \text{Compute 1, remove 1.} \]
Signature-based criteria

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

Sketch of proof

1. \( s(\alpha - \beta) < s(\alpha), s(\beta) \).
2. All S-pairs are handled by increasing signature.
   \( \Rightarrow \) All relations \( < s(\alpha) \) are known:
   \[ \alpha = \beta + \text{elements of smaller signature} \]
Signature-based criteria

S-pairs in signature $T$
Signature-based criteria

S-pairs in signature $T$

What are all possible configurations to reach signature $T$?
Signature-based criteria

S-pairs in signature $T$

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

What are all possible configurations to reach signature $T$?
Signature-based criteria

What are all possible configurations to reach signature $T$?

Define an order on $R_T$ and choose the maximal element.

$R_T = \{ a\alpha \mid \alpha \text{ handled by the algorithm and } s(a\alpha) = T \}$
Special cases

\[ \mathcal{R}_T = \left\{ a\alpha \mid \alpha \text{ handled by the algorithm and } s(a\alpha) = T \right\} \]
Special cases

\[ \mathcal{R}_T = \{ a\alpha \mid \alpha \text{ handled by the algorithm and } s(a\alpha) = T \} \]

Choose \( b\beta \) to be an element of \( \mathcal{R}_T \) maximal w.r.t. an order \( \leq \).
Special cases

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

Choose $b\beta$ to be an element of $\mathcal{R}_T$ maximal w.r.t. an order $\preceq$.

1. **If $b\beta$ is a syzygy** $\implies$ Go on to next signature.
Special cases

\[ \mathcal{R}_T = \{ a\alpha \mid \alpha \text{ handled by the algorithm and } s(a\alpha) = T \} \]

Choose \( b\beta \) to be an element of \( \mathcal{R}_T \) maximal w.r.t. an order \( \preceq \).

1. If \( b\beta \) is a syzygy \( \implies \) Go on to next signature.
2. If \( b\beta \) is not part of an S-pair \( \implies \) Go on to next signature.
Special cases

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

Choose \( b\beta \) to be an element of \( \mathcal{R}_T \) maximal w.r.t. an order \( \sqsubseteq \).

1. If \( b\beta \) is a syzygy \( \implies \) Go on to next signature.
2. If \( b\beta \) is not part of an S-pair \( \implies \) Go on to next signature.

Revisiting our example with \( \prec_{\text{pot}} \)

\[ s(\text{spol}(g_3, g_1)) = xye_2 \]

\[ \begin{aligned} g_1 &= xy - z^2 \\ g_2 &= y^2 - z^2 \end{aligned} \]

\[ \Rightarrow \text{psyz}(g_2, g_1) = g_1 e_2 - g_2 e_1 = xye_2 + \ldots \]
# zero reductions (Singular-4-0-0, $\mathbb{F}_{32003}$)

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>STD</th>
<th>SBA $\prec_{pot}$</th>
<th>SBA $\prec_{d-pot}$</th>
<th>SBA $\prec_{lt}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>cyclic-8</td>
<td>4,284</td>
<td>243</td>
<td>243</td>
<td>671</td>
</tr>
<tr>
<td>cyclic-8-h</td>
<td>5,843</td>
<td>243</td>
<td>243</td>
<td>671</td>
</tr>
<tr>
<td>eco-11</td>
<td>3,476</td>
<td>0</td>
<td>749</td>
<td>749</td>
</tr>
<tr>
<td>eco-11-h</td>
<td>5,429</td>
<td>502</td>
<td>502</td>
<td>749</td>
</tr>
<tr>
<td>katsura-11</td>
<td>3,933</td>
<td>0</td>
<td>0</td>
<td>348</td>
</tr>
<tr>
<td>katsura-11-h</td>
<td>3,933</td>
<td>0</td>
<td>0</td>
<td>348</td>
</tr>
<tr>
<td>noon-9</td>
<td>25,508</td>
<td>0</td>
<td>0</td>
<td>682</td>
</tr>
<tr>
<td>noon-9-h</td>
<td>25,508</td>
<td>0</td>
<td>0</td>
<td>682</td>
</tr>
<tr>
<td>Random(11,2,2)</td>
<td>6,292</td>
<td>0</td>
<td>0</td>
<td>590</td>
</tr>
<tr>
<td>HRandom(11,2,2)</td>
<td>6,292</td>
<td>0</td>
<td>0</td>
<td>590</td>
</tr>
<tr>
<td>Random(12,2,2)</td>
<td>13,576</td>
<td>0</td>
<td>0</td>
<td>1,083</td>
</tr>
<tr>
<td>HRandom(12,2,2)</td>
<td>13,576</td>
<td>0</td>
<td>0</td>
<td>1,083</td>
</tr>
</tbody>
</table>
### Time in seconds (Singular-4-0-0, $\mathbb{F}_{32003}$)

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>STD</th>
<th>SBA $\prec_{\text{pot}}$</th>
<th>SBA $\prec_{d\text{-pot}}$</th>
<th>SBA $\prec_{\text{lt}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>cyclic-8</td>
<td>32.480</td>
<td>44.310</td>
<td>100.780</td>
<td>31.120</td>
</tr>
<tr>
<td>cyclic-8-h</td>
<td>38.300</td>
<td>35.770</td>
<td>98.440</td>
<td>32.640</td>
</tr>
<tr>
<td>eco-11</td>
<td>28.450</td>
<td>3.450</td>
<td>27.360</td>
<td>13.270</td>
</tr>
<tr>
<td>eco-11-h</td>
<td>20.630</td>
<td>11.600</td>
<td>14.840</td>
<td>7.960</td>
</tr>
<tr>
<td>katsura-11</td>
<td>54.780</td>
<td>35.720</td>
<td>31.010</td>
<td>11.790</td>
</tr>
<tr>
<td>katsura-11-h</td>
<td>51.260</td>
<td>34.080</td>
<td>32.590</td>
<td>17.230</td>
</tr>
<tr>
<td>noon-9</td>
<td>29.730</td>
<td>12.940</td>
<td>14.620</td>
<td>15.220</td>
</tr>
<tr>
<td>noon-9-h</td>
<td>34.410</td>
<td>17.850</td>
<td>20.090</td>
<td>20.510</td>
</tr>
<tr>
<td>Random(11,2,2)</td>
<td>267.810</td>
<td>77.430</td>
<td>130.400</td>
<td>28.640</td>
</tr>
<tr>
<td>HRandom(11,2,2)</td>
<td>22.970</td>
<td>14.060</td>
<td>39.320</td>
<td>3.540</td>
</tr>
<tr>
<td>Random(12,2,2)</td>
<td>2,069.890</td>
<td>537.340</td>
<td>1,062.390</td>
<td>176.920</td>
</tr>
<tr>
<td>HRandom(12,2,2)</td>
<td>172.910</td>
<td>112.420</td>
<td>331.680</td>
<td>22.060</td>
</tr>
</tbody>
</table>
Predicting zero reductions

Fast linear algebra for computing Gröbner bases
Faugère’s F4 algorithm

Input: Ideal \( I = \langle f_1, \ldots, f_m \rangle \)
Output: Gröbner basis \( G \) for \( I \)

1. \( G \leftarrow \emptyset \)
2. \( G \leftarrow G \cup \{ f_i \} \) for all \( i \in \{1, \ldots, m\} \)
3. Set \( P \leftarrow \{(af, bg) \mid f, g \in G\} \)
4. \( d \leftarrow 0 \)
5. while \( P \neq \emptyset \):
   ▶ If \( \text{lt}(h) \notin L(G) \) (all other \( h \) are "useless"):
      \( \leftarrow \leftarrow \emptyset \)
      \( \leftarrow \leftarrow G \cup \{ h \} \)
Faugère’s F4 algorithm

**Input:** Ideal \( I = \langle f_1, \ldots, f_m \rangle \)

**Output:** Gröbner basis \( G \) for \( I \)

1. \( G \leftarrow \emptyset \)
2. \( G \leftarrow G \cup \{ f_i \} \) for all \( i \in \{1, \ldots, m\} \)
3. Set \( P \leftarrow \{ (af, bg) \mid f, g \in G \} \)
4. \( d \leftarrow 0 \)
5. while \( P \neq \emptyset \):
   (a) \( d \leftarrow d + 1 \)
   (b) \( P_d \leftarrow \textbf{Select}(P) \), \( P \leftarrow P \setminus P_d \)
   (c) \( L_d \leftarrow \{ af, bg \mid (af, bg) \in P_d \} \)
   (d) \( L_d \leftarrow \textbf{Symbolic Preprocessing}(L_d, G) \)
   (e) \( F_d \leftarrow \textbf{Reduction}(L_d, G) \)
   (f) for \( h \in F_d \):
      ▶ If \( \text{lt}(h) \notin L(G) \) (all other \( h \) are “useless”):
         ▶ \( P \leftarrow P \cup \{ \text{new pairs with } h \} \)
         ▶ \( G \leftarrow G \cup \{ h \} \)
6. Return \( G \)
Differences to Buchberger

1. **Select a subset** $P_d$ of $P$, not only one element.
2. Do a **symbolic preprocessing**:
   - Search and store reducers, but do not reduce.
3. Do a **full reduction of** $P_d$ at once:
   - Reduce a subset of $\mathcal{R}$ by a subset of $\mathcal{R}$
Differences to Buchberger

1. **Select a subset** $P_d$ of $P$, not only one element.

2. Do a **symbolic preprocessing**:  
   Search and store reducers, but do not reduce.

3. Do a **full reduction of** $P_d$ at once:  
   Reduce a subset of $R$ by a subset of $R$

If **Select** $(P)$ selects only one pair F4 is just Buchberger’s algorithm.  
Usually one chooses the normal selection strategy,  
i.e. all pairs of lowest degree.
Symbolic preprocessing

Input: $L, G$ finite subsets of $\mathbb{R}$
Output: a finite subset of $\mathbb{R}$

1. $F \leftarrow L$
2. $D \leftarrow L(F)$ (S-pairs already reduce lead terms)
3. while $T(F) \neq D$:
   (a) Choose $m \in T(F) \setminus D$, $D \leftarrow D \cup \{m\}$.
   (b) If $m \in L(G) \Rightarrow \exists g \in G$ and $\lambda \in \mathbb{R}$ such that $\lambda \text{lt}(g) = m$
     $\Rightarrow F \leftarrow F \cup \{\lambda g\}$
4. Return $F$
Symbolic preprocessing

**Input:** \( L, G \) finite subsets of \( \mathbb{R} \)

**Output:** a finite subset of \( \mathbb{R} \)

1. \( F \leftarrow L \)
2. \( D \leftarrow L(F) \) \((S\text{-pairs already reduce lead terms})\)
3. while \( T(F) \neq D \):
   (a) Choose \( m \in T(F) \setminus D \), \( D \leftarrow D \cup \{m\} \).
   (b) If \( m \in L(G) \Rightarrow \exists g \in G \) and \( \lambda \in \mathbb{R} \) such that \( \lambda \text{lt}(g) = m \)
      \( \triangleright F \leftarrow F \cup \{\lambda g\} \)
4. Return \( F \)

We optimize this soon!
Reduction

Input: $L$ finite subsets of $\mathbb{R}$
Output: a finite subset of $\mathbb{R}$

1. $M \leftarrow$ Macaulay matrix of $L$
2. $M \leftarrow$ Gaussian Elimination of $M$ (Linear algebra)
3. $F \leftarrow$ polynomials from rows of $M$
4. Return $F$
Input: $L$ finite subsets of $\mathbb{R}$
Output: a finite subset of $\mathbb{R}$

1. $M \leftarrow$ Macaulay matrix of $L$
2. $M \leftarrow$ Gaussian Elimination of $M$ (Linear algebra)
3. $F \leftarrow$ polynomials from rows of $M$
4. Return $F$

**Macaulay matrix**
- columns $\triangleright$ monomials (sorted by monomial order $\prec$)
- rows $\triangleright$ coefficients of polynomials in $L$
Example: Cyclic-4

\[ R = \mathbb{Q}[a, b, c, d], \quad < \text{denotes DRL and we use the normal selection strategy for} \ Select(P). \ i = \langle f_1, \ldots, f_4 \rangle, \text{where} \]

\begin{align*}
    f_1 &= abcd - 1, \\
    f_2 &= abc + abd + acd + bcd, \\
    f_3 &= ab + bc + ad + cd, \\
    f_4 &= a + b + c + d.
\end{align*}
\( \mathcal{R} = \mathbb{Q}[a, b, c, d] \), \( < \) denotes DRL and we use the normal selection strategy for \textbf{Select}(P). \( l = \langle f_1, \ldots, f_4 \rangle \), where

\[
\begin{align*}
  f_1 &= abcd - 1, \\
  f_2 &= abc + abd + acd + bcd, \\
  f_3 &= ab + bc + ad + cd, \\
  f_4 &= a + b + c + d.
\end{align*}
\]

We start with \( G = \{ f_4 \} \) and \( P_1 = \{(f_3, bf_4)\} \), thus \( L_1 = \{ f_3, bf_4 \} \).
Example: Cyclic-4

\( \mathcal{R} = \mathbb{Q}[a, b, c, d], \) denotes DRL and we use the normal selection strategy for \textbf{Select}(P). \( I = \langle f_1, \ldots, f_4 \rangle, \) where

\[
\begin{align*}
  f_1 &= abcd - 1, \\
  f_2 &= abc + abd + acd + bcd, \\
  f_3 &= ab + bc + ad + cd, \\
  f_4 &= a + b + c + d.
\end{align*}
\]

We start with \( G = \{ f_4 \} \) and \( P_1 = \{ (f_3, bf_4) \} \), thus \( L_1 = \{ f_3, bf_4 \} \).

Let us do symbolic preprocessing:

\[
T(L_1) = \{ ab, b^2, bc, ad, bd, cd \}
\]
\[
L_1 = \{ f_3, bf_4 \}
\]
Example: Cyclic-4

\[ \mathcal{R} = \mathbb{Q}[a, b, c, d], < \text{ denotes DRL and we use the normal selection strategy for } \textbf{Select}(P). \ i = \langle f_1, \ldots, f_4 \rangle, \text{ where} \]

\[
\begin{align*}
    f_1 &= abcd - 1, \\
    f_2 &= abc + abd + acd + bcd, \\
    f_3 &= ab + bc + ad + cd, \\
    f_4 &= a + b + c + d.
\end{align*}
\]

We start with \( G = \{ f_4 \} \) and \( P_1 = \{ (f_3, bf_4) \} \), thus \( L_1 = \{ f_3, bf_4 \} \).

Let us do symbolic preprocessing:

\[
\begin{align*}
    T(L_1) &= \{ ab, b^2, bc, ad, bd, cd \} \\
    L_1 &= \{ f_3, bf_4 \}
\end{align*}
\]

\( b^2 \notin L(G) \),
Example: Cyclic-4

\( \mathcal{R} = \mathbb{Q}[a, b, c, d] \), \( \prec \) denotes DRL and we use the normal selection strategy for \textbf{Select}(\( P \)). \( I = \langle f_1, \ldots, f_4 \rangle \), where

\[
\begin{align*}
f_1 &= abcd - 1, \\
f_2 &= abc + abd + acd + bcd, \\
f_3 &= ab + bc + ad + cd, \\
f_4 &= a + b + c + d.
\end{align*}
\]

We start with \( G = \{ f_4 \} \) and \( P_1 = \{ (f_3, bf_4) \} \), thus \( L_1 = \{ f_3, bf_4 \} \).

Let us do symbolic preprocessing:

\[
\begin{align*}
T(L_1) &= \{ ab, b^2, bc, ad, bd, cd \} \\
L_1 &= \{ f_3, bf_4 \}
\end{align*}
\]

\( b^2 \notin L(G) \), \( bc \notin L(G) \),
Example: Cyclic-4

\( R = \mathbb{Q}[a, b, c, d], < \text{ denotes DRL and we use the normal selection strategy for } \textbf{Select}(P). \ l = \langle f_1, \ldots, f_4 \rangle, \text{ where} \)

\[
\begin{align*}
  f_1 &= abcd - 1, \\
  f_2 &= abc + abd + acd + bcd, \\
  f_3 &= ab + bc + ad + cd, \\
  f_4 &= a + b + c + d.
\end{align*}
\]

We start with \( G = \{ f_4 \} \) and \( P_1 = \{ (f_3, bf_4) \} \), thus \( L_1 = \{ f_3, bf_4 \} \).

Let us do symbolic preprocessing:

\[
\begin{align*}
  T(L_1) &= \{ ab, b^2, bc, ad, bd, cd, d^2 \} \\
  L_1 &= \{ f_3, bf_4, df_4 \}
\end{align*}
\]

\( b^2 \notin L(G), \ bc \notin L(G), \ d\text{lt}(f_4) = ad, \)
Example: Cyclic-4

\( \mathcal{R} = \mathbb{Q}[a, b, c, d] \), \( < \) denotes DRL and we use the normal selection strategy for \textbf{Select}(\( P \)). \( I = \langle f_1, \ldots, f_4 \rangle \), where

\[
\begin{align*}
  f_1 &= abcd - 1, \\
  f_2 &= abc + abd + acd + bcd, \\
  f_3 &= ab + bc + ad + cd, \\
  f_4 &= a + b + c + d.
\end{align*}
\]

We start with \( G = \{f_4\} \) and \( P_1 = \{(f_3, bf_4)\} \), thus \( L_1 = \{f_3, bf_4\} \).

Let us do symbolic preprocessing:

\[
\begin{align*}
  T(L_1) &= \{ab, b^2, bc, ad, bd, cd, d^2\} \\
  L_1 &= \{f_3, bf_4, df_4\}
\end{align*}
\]

\( b^2 \not\in L(G), \ bc \not\in L(G), \ df_4(f_4) = ad \), all others also \( \not\in L(G), \)
Example: Cyclic-4

Now reduction:
Convert polynomial data $L_1$ to Macaulay Matrix $M_1$

\[
\begin{pmatrix}
ab & b^2 & bc & ad & bd & cd & d^2 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 0
\end{pmatrix}
\]
Example: Cyclic-4

Now reduction:
Convert polynomial data $L_1$ to Macaulay Matrix $M_1$

$$
\begin{align*}
\text{df}_4 & \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ df_4 & \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\end{align*}
\end{align*}
$$

Gaussian Elimination of $M_1$:

$$
\begin{align*}
\text{df}_4 & \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ f_3 & \begin{pmatrix} 1 & 0 & 1 & 0 & -1 & 0 & -1 \\
bf_4 & \begin{pmatrix} 0 & 1 & 0 & 0 & 2 & 0 & 1 \\
\end{align*}
\end{align*}
$$
Example: Cyclic-4

Convert matrix data back to polynomial structure $F_1$:

$$
\begin{bmatrix}
ab & b^2 & bc & ad & bd & cd & d^2 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & -1 & 0 & -1 \\
0 & 1 & 0 & 0 & 2 & 0 & 1
\end{bmatrix}
$$

\[
F_1 = \left\{ \begin{array}{l}
\underbrace{ad + bd + cd + d^2}_{f_5},
\underbrace{ab + bc - bd - d^2}_{f_6},
\underbrace{b^2 + 2bd + d^2}_{f_7}
\end{array} \right\}
\]
Example: Cyclic-4

Convert matrix data back to polynomial structure $F_1$:

$$
\begin{pmatrix}
ab & b^2 & bc & ad & bd & cd & d^2 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & -1 & 0 & -1 \\
0 & 1 & 0 & 0 & 2 & 0 & 1
\end{pmatrix}
$$

$$
F_1 = \begin{cases}
ad + bd + cd + d^2, & f_5 \\
ab + bc - bd - d^2, & f_6 \\
b^2 + 2bd + d^2, & f_7
\end{cases}
$$

\[\text{lt}(f_5), \text{lt}(f_6) \in L(G), \text{ so} \]

\[G \leftarrow G \cup \{f_7\}.\]
Next round:

\[ G = \{ f_4, f_7 \}, \ P_2 = \{ (f_2, bcf_4) \}, \ L_2 = \{ f_2, bcf_4 \}. \]
Next round:

\[ G = \{ f_4, f_7 \}, \quad P_2 = \{ (f_2, bcf_4) \}, \quad L_2 = \{ f_2, bcf_4 \}. \]

We can simplify the computations:

\[ \text{lt} (bcf_4) = abc = \text{lt} (cf_6). \]

\( f_6 \) possibly better reduced than \( f_4 \). (\( f_6 \) is not in \( G \! \)!)
Example: Cyclic-4

Next round:

\[ G = \{ f_4, f_7 \}, \quad P_2 = \{(f_2, bcf_4)\}, \quad L_2 = \{ f_2, bcf_4 \}. \]

We can simplify the computations:

\[ \text{lt}(bcf_4) = abc = \text{lt}(cf_6). \]

\( f_6 \) possibly better reduced than \( f_4 \). (\( f_6 \) is not in \( G \)!

\[ \implies L_2 = \{ f_2, cf_6 \} \]

Symbolic preprocessing:

\[ T(L_2) = \{ abc, bc^2, abd, acd, bcd, cd^2 \} \]

\[ L_2 = \{ f_2, cf_6 \} \]
Example: Cyclic-4

Next round:

\[ G = \{ f_4, f_7 \}, \quad P_2 = \{(f_2, bcf_4)\}, \quad L_2 = \{ f_2, bcf_4 \}. \]

We can simplify the computations:

\[ \text{lt}(bcf_4) = abc = \text{lt}(cf_6). \]

\( f_6 \) possibly better reduced than \( f_4 \). (\( f_6 \) is not in \( G \! \)!) \[ \implies L_2 = \{ f_2, cf_6 \} \]

Symbolic preprocessing:

\[ T(L_2) = \{ abc, bc^2, abd, acd, bcd, cd^2 \} \]

\[ L_2 = \{ f_2, cf_6, \} \]

\( bc^2 \notin L(G), \)
Example: Cyclic-4

Next round:

\[ G = \{f_4, f_7\}, \quad P_2 = \{(f_2, bcf_4)\}, \quad L_2 = \{f_2, bcf_4\}. \]

We can simplify the computations:

\[ \text{lt}(bcf_4) = abc = \text{lt}(cf_6). \]

\(f_6\) possibly better reduced than \(f_4\). \((f_6\ \text{is not in} \ G!\)

\[ \Rightarrow \quad L_2 = \{f_2, cf_6\}\]

Symbolic preprocessing:

\[ T(L_2) = \{abc, bc^2, abd, acd, bcd, cd^2\} \]

\[ L_2 = \{f_2, cf_6\} \]

\(bc^2 \not\in L(G), \quad abd = \text{lt}(bdf_4), \quad \text{but also} \quad abd = \text{lt}(bf_5)! \)
Next round:

\[ G = \{ f_4, f_7 \}, \; P_2 = \{(f_2, bcf_4)\}, \; L_2 = \{ f_2, bcf_4 \}. \]

We can simplify the computations:

\[ \text{lt}(bcf_4) = abc = \text{lt}(cf_6). \]

\( f_6 \) possibly better reduced than \( f_4 \). (\( f_6 \) is not in \( G \! \))

\[ \implies L_2 = \{ f_2, cf_6 \} \]

Symbolic preprocessing:

\[ T(L_2) = \{ abc, bc^2, abd, acd, bcd, cd^2 \} \]

\[ L_2 = \{ f_2, cf_6 \} \]

\( bc^2 \notin L(G), \; abd = \text{lt}(bdf_4) \), but also \( abd = \text{lt}(bf_5) \! \)

Let us investigate this in more detail.
Interlude – Simplify

**Idea**
Replace $u \cdot f$ by $(wv) \cdot g$ where $vg \in F_i$ for a previous reduction step.  
$\Rightarrow$ Reuse rows that are reduced but not “in” $G$.  

$\text{Note}$
$\Rightarrow$ Tries to reuse all rows from old matrices.  
$\Rightarrow$ We need to keep them in memory.  
$\Rightarrow$ We also simplify generators of $S$-pairs, as we have done in our example: $(f_2, bfc_4) = (f_2, cf_6)$.  
$\Rightarrow$ One can also choose “better” reducers by other properties, not only “last reduced one”.  
$\Rightarrow$ Without Simplify the $F4$ algorithm is rather slow.  
In our example: Choose $bf_5$ as reducer, not $bdf_4$.  

Interlude – Simplify

**Idea**
Replace $u \cdot f$ by $(wv) \cdot g$ where $vg \in F_i$ for a previous reduction step.
⇒ Reuse rows that are reduced but not “in” $G$.

**Note**
- Tries to reuse all rows from old matrices.
  ⇒ We need to keep them in memory.
- We also simplify generators of S-pairs, as we have done in our example: $(f_2, bcf_4) \rightarrow (f_2, cf_6)$.
- One can also choose “better” reducers by other properties, not only “last reduced one”.
- Without **Simplify** the F4 algorithm is rather slow.
Interlude – Simplify

**Idea**
Replace $u \cdot f$ by $(wv) \cdot g$ where $vg \in F_i$ for a previous reduction step.
⇒ Reuse rows that are reduced but not “in” $G$.

**Note**

- Tries to reuse all rows from old matrices.
  ⇒ We need to keep them in memory.
- We also simplify generators of S-pairs, as we have done in our example: $(f_2, bcf_4) \implies (f_2, cf_6)$.
- One can also choose “better” reducers by other properties, not only “last reduced one”.
- Without **Simplify** the F4 algorithm is rather slow.

In our example:
Choose $bf_5$ as reducer, not $bdf_4$. 
Example: Cyclic-4

Symbolic preprocessing - now with **simplify**:

\[
T(L_2) = \{ abc, bc^2, abd, acd, bcd, cd^2 \}
\]

\[
L_2 = \{ f_2, cf_6 \}
\]

\[ bc^2 \notin L(G), \]
Example: Cyclic-4

Symbolic preprocessing - now with simplify:

\[ T(L_2) = \{ abc, bc^2, abd, acd, bcd, cd^2 \} \]
\[ L_2 = \{ f_2, cf_6 \} \]

\( bc^2 \notin L(G), \quad abd = \text{lt}(bf_5), \)
Symbolic preprocessing - now with \textit{simplify}:

\[
T(L_2) = \{ abc, bc^2, abd, acd, bcd, cd^2, b^2d, c^2d \}
\]
\[
L_2 = \{ f_2, cf_6, bf_5 \}
\]

\( bc^2 \notin L(G), \ abd = \text{lt}(bf_5), \)
Symbolic preprocessing - now with simplify:

\[
T(L_2) = \{ abc, bc^2, abd, acd, bcd, cd^2, b^2d, c^2d, \ldots \}
\]

\[
L_2 = \{ f_2, cf_6, bf_5, cf_5, df_7 \}
\]

\(bc^2 \notin L(G), \ abd = \text{lt}(bf_5), \) and so on.
Example: Cyclic-4

Symbolic preprocessing - now with simplify:

\[ T(L_2) = \{ abc, bc^2, abd, acd, bcd, cd^2, b^2d, c^2d, \ldots \} \]

\[ L_2 = \{ f_2, cf_6, bf_5, cf_5, df_7 \} \]

\( bc^2 \notin L(G), \ abd = \text{lt}(bf_5) \), and so on.

Now try to exploit the special structure of the Macaulay matrices.
Improve Gaussian Elimination

Use **Linear Algebra** for reduction steps in GB computations.
Improve Gaussian Elimination

Use **Linear Algebra** for reduction steps in GB computations.

\[
\begin{array}{cccccccc}
1 & 3 & 0 & 0 & 7 & 1 & 0 \\
1 & 0 & 4 & 1 & 0 & 0 & 5 \\
0 & 1 & 6 & 0 & 8 & 0 & 1 \\
0 & 5 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 1 & 3 & 1 \\
\end{array}
\]
Improve Gaussian Elimination

Use **Linear Algebra** for reduction steps in GB computations.

\[
\begin{bmatrix}
1 & 3 & 0 & 0 & 7 & 1 & 0 \\
1 & 0 & 4 & 1 & 0 & 0 & 5 \\
0 & 1 & 6 & 0 & 8 & 0 & 1 \\
0 & 5 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 1 & 3 & 1
\end{bmatrix}
\]

Knowledge of underlying GB structure
Improve Gaussian Elimination

Use **Linear Algebra** for reduction steps in GB computations.

\[
\begin{align*}
\text{S-pair} & \quad \begin{cases} 
1 & 3 & 0 & 0 & 7 & 1 & 0 \\
1 & 0 & 4 & 1 & 0 & 0 & 5 \\
0 & 1 & 6 & 0 & 8 & 0 & 1 \\
0 & 5 & 0 & 0 & 0 & 2 & 0 \\
\end{cases} \\
\text{reducer} & \quad \begin{cases} 
0 & 0 & 0 & 0 & 1 & 3 & 1 \\
\end{cases}
\end{align*}
\]

Knowledge of underlying GB structure
Improve Gaussian Elimination

Use **Linear Algebra** for reduction steps in GB computations.

\[
\begin{align*}
\text{S-pair} & : 
\begin{cases}
1 & 3 & 0 & 0 & 7 & 1 & 0 \\
1 & 0 & 4 & 1 & 0 & 0 & 5 \\
0 & 1 & 6 & 0 & 8 & 0 & 1 \\
0 & 5 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 1 & 3 & 1
\end{cases} \\
\text{reducer} & : 0 & 0 & 0 & 0 & 1 & 3 & 1
\end{align*}
\]

Knowledge of underlying GB structure
Improve Gaussian Elimination

Use **Linear Algebra** for reduction steps in GB computations.

\[
\begin{align*}
\text{S-pair} & \quad \begin{cases} 
1 & 3 & 0 & 0 & 7 & 1 & 0 \\
1 & 0 & 4 & 1 & 0 & 0 & 5 \\
0 & 1 & 6 & 0 & 8 & 0 & 1 \\
0 & 5 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 1 & 3 & 1 
\end{cases} \\
\text{reducer} & \quad \leftarrow
\end{align*}
\]

Knowledge of underlying GB structure

**Idea**

Do a static **reordering before** the Gaussian Elimination to achieve a better initial shape. **Reorder afterwards.**
Faugère-Lachartre Idea

1st step: Sort pivot and non-pivot columns

```
1 3 0 0 7 1 0
1 0 4 1 0 0 5
0 1 6 0 8 0 1
0 5 0 0 0 2 0
0 0 0 0 1 3 1
```
Faugère-Lachartre Idea

1st step: Sort pivot and non-pivot columns

1 3 0 0 7 1 0
1 0 4 1 0 0 5
0 1 6 0 8 0 1
0 5 0 0 0 2 0
0 0 0 0 1 3 1

Pivot column
**Faugère-Lachartre Idea**

**1st step**: Sort pivot and non-pivot columns

<table>
<thead>
<tr>
<th>1</th>
<th>3</th>
<th>0</th>
<th>0</th>
<th>7</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Pivot column
1st step: Sort pivot and non-pivot columns

<table>
<thead>
<tr>
<th>Pivot column</th>
<th>Non-Pivot column</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 3 0 0 7 1 0</td>
<td></td>
</tr>
<tr>
<td>1 0 4 1 0 0 5</td>
<td></td>
</tr>
<tr>
<td>0 1 6 0 8 0 1</td>
<td></td>
</tr>
<tr>
<td>0 5 0 0 0 2 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 1 3 1</td>
<td></td>
</tr>
</tbody>
</table>
1st step: Sort pivot and non-pivot columns

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Pivot column

Non-Pivot column
**Faugère-Lachartre Idea**

**1st step**: Sort pivot and non-pivot columns

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>3</th>
<th>0</th>
<th>0</th>
<th>7</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pivot column</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Non-Pivot column</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>3</th>
<th>7</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pivot column</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Non-Pivot column</td>
<td>0</td>
<td>1</td>
<td>8</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
2nd step: Sort pivot and non-pivot rows

\[
\begin{array}{cccccc}
1 & 3 & 7 & 0 & 0 & 1 \\
1 & 0 & 0 & 4 & 1 & 0 \\
1 & 0 & 0 & 4 & 1 & 0 \\
0 & 1 & 8 & 6 & 0 & 0 \\
0 & 5 & 0 & 0 & 0 & 2 \\
0 & 0 & 1 & 0 & 0 & 3 \\
0 & 0 & 1 & 0 & 0 & 3 \\
\end{array}
\]
2nd step: Sort pivot and non-pivot rows

1 3 7 0 0 1 0
1 0 0 4 1 0 5
0 1 8 6 0 0 9
0 5 0 0 0 2 0
0 0 1 0 0 3 1

Pivot row
Faugère-Lachartre Idea

**2nd step**: Sort pivot and non-pivot rows

\[
\begin{array}{cccccccc}
1 & 3 & 7 & 0 & 0 & 1 & 0 & \text{Pivot row} \\
1 & 0 & 0 & 4 & 1 & 0 & 5 & \text{Non-Pivot row} \\
0 & 1 & 8 & 6 & 0 & 0 & 9 & \\
0 & 5 & 0 & 0 & 0 & 2 & 0 & \\
0 & 0 & 1 & 0 & 0 & 3 & 1 & \\
\end{array}
\]
Faugère-Lachartre Idea

2nd step: Sort pivot and non-pivot rows

```
1 3 7 0 0 1 0
1 0 0 4 1 0 5
0 1 8 6 0 0 9
0 5 0 0 0 2 0
0 0 1 0 0 3 1
```

Pivot row

Non-Pivot row
**Faugère-Lachartre Idea**

**2nd step**: Sort pivot and non-pivot rows

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>3</th>
<th>7</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>8</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>4</th>
<th>1</th>
<th>0</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>8</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>9</td>
</tr>
</tbody>
</table>
3rd step: Reduce lower left part to zero

\[
\begin{array}{cccc}
1 & 0 & 0 & 4 \\
0 & 5 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 3 & 7 & 0 \\
0 & 1 & 8 & 6 \\
0 & 0 & 1 & 0 \\
0 & 0 & 3 & 1 \\
0 & 0 & 0 & 9
\end{array}
\]
Faugère-Lachartre Idea

**3rd step**: Reduce lower left part to zero

```
| 1 0 0 | 4 1 0 5 |
| 0 5 0 | 0 0 2 0 |
| 0 0 1 | 0 0 3 1 |
| 1 3 7 | 0 0 1 0 |
| 0 1 8 | 6 0 0 9 |
```

```
| 1 0 0 | 4 1 0 5 |
| 0 5 0 | 0 0 2 0 |
| 0 0 1 | 0 0 3 1 |
| 0 0 0 | 7 10 3 10 |
| 0 0 0 | 6 0 2 1 |
```
Faugère-Lachartre Idea

**4th step**: Reduce lower right part

```
1 0 0 4 1 0 5
0 5 0 0 0 2 0
0 0 1 0 0 3 1
0 0 0 7 10 3 10
0 0 0 6 0 2 1
```
**Faugère-Lachartre Idea**

**4th step:** Reduce lower right part

```
1 0 0 4 1 0 5
0 5 0 0 0 2 0
0 0 1 0 0 3 1
0 0 0 7 10 3 10
0 0 0 6 0 2 1
```

→

```
1 0 0 4 1 0 5
0 5 0 0 0 2 0
0 0 1 0 0 3 1
0 0 0 7 10 3 10
0 0 0 0 4 1 5
```
Faugère-Lachartre Idea

**4th step**: Reduce lower right part

\[
\begin{array}{cccc}
1 & 0 & 0 & 4 \\
0 & 5 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 7 \\
0 & 0 & 0 & 6 \\
\end{array}
\quad\quad\quad
\begin{array}{cccc}
1 & 0 & 0 & 4 \\
0 & 5 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 7 \\
0 & 0 & 0 & 6 \\
\end{array}
\]

**5th step**: Remap columns of lower right part

\[
\begin{array}{cccc}
1 & 0 & 0 & 5 \\
0 & 5 & 0 & 2 \\
0 & 0 & 3 & 1 \\
0 & 7 & 10 & 3 \\
0 & 6 & 2 & 1 \\
\end{array}
\quad\quad\quad
\begin{array}{cccc}
1 & 0 & 0 & 5 \\
0 & 5 & 0 & 2 \\
0 & 0 & 3 & 1 \\
0 & 7 & 10 & 3 \\
0 & 6 & 2 & 1 \\
\end{array}
\]
How our matrices look like (1)
How our matrices look like (2)
Hybrid Matrix Multiplication $A^{-1}B$
Hybrid Matrix Multiplication $A^{-1}B$
Reduce $C$ to zero
Gaussian Elimination on $D$
New information
GBLA

- New open-source, plain C library, specialized linear algebra for GB computations
- at the moment: dedicated to finite fields, $p \leq 65521 < 2^{16}$
- written together with Brice Boyer and Jean-Charles Faugère
- several strategies for splicing and reduction steps
- includes converter for our dedicated matrix format, e.g. from/to Magma
- comes with a huge matrix database, > 280 GB of data
New open-source, plain C library, specialized linear algebra for GB computations

at the moment: dedicated to finite fields, $p \leq 65521 < 2^{16}$

written together with Brice Boyer and Jean-Charles Faugère

several strategies for splicing and reduction steps

includes converter for our dedicated matrix format, e.g. from/to Magma

comes with a huge matrix database, $> 280$ GB of data

Repository will soon be open for external contributions!

http://hpac.imag.fr/gbla/
### GBLA vs. Faugère-Lachartre

<table>
<thead>
<tr>
<th>Implementation</th>
<th>FL reduction</th>
<th>GBLA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix/Threads:</td>
<td>1 16 32</td>
<td>1 16 32</td>
</tr>
<tr>
<td>$F_5$ kat13 mat5</td>
<td>16.7 2.7 2.3</td>
<td>14.5 2.02 1.87</td>
</tr>
<tr>
<td></td>
<td>mat6</td>
<td>27.3 4.15 4.0</td>
</tr>
<tr>
<td>$F_5$ kat14 mat7</td>
<td>139 17.4 16.6</td>
<td>142 13.4 10.6</td>
</tr>
<tr>
<td></td>
<td>mat8</td>
<td>181 24.95 23.1</td>
</tr>
<tr>
<td>$F_5$ kat15 mat7</td>
<td>629 61.8 55.6</td>
<td>633 55.1 38.2</td>
</tr>
<tr>
<td>$F_5$ kat16 mat6</td>
<td>1,203 110 83.3</td>
<td>1,147 98.7 69.9</td>
</tr>
<tr>
<td>$F_5$ mr-9-10-7 mat3</td>
<td>591 70.8 71.3</td>
<td>733 57.3 37.9</td>
</tr>
</tbody>
</table>
## GBLA vs. Magma V2.20-10

<table>
<thead>
<tr>
<th>Implementation</th>
<th>Magma</th>
<th>GBLA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Matrix/Threads:</strong></td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td><strong>$F_4$</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kat12 mat9</td>
<td>11.2</td>
<td>11.4</td>
</tr>
<tr>
<td>kat13 mat2</td>
<td>0.94</td>
<td>1.18</td>
</tr>
<tr>
<td>mat3</td>
<td>9.33</td>
<td>11.0</td>
</tr>
<tr>
<td>mat9</td>
<td>168</td>
<td>165</td>
</tr>
<tr>
<td>mat8</td>
<td>2747</td>
<td>2545</td>
</tr>
<tr>
<td>mat7</td>
<td>10,345</td>
<td>9,514</td>
</tr>
<tr>
<td>mat8</td>
<td>13,936</td>
<td>12,547</td>
</tr>
<tr>
<td>mat9</td>
<td>24,393</td>
<td>22,247</td>
</tr>
</tbody>
</table>


