An Introduction to F4, some remarks on F5

Christian Eder

University of Kaiserslautern

September 15, 2010

Connections between Buchberger's Algorithm and Faugère's F4

Similarities

1. Both compute a Gröbner basis G for a finite set of polynomials F.
2. Both generate pairs of elements of the input, reduce these, add newly generated elements to G, generate new pairs of elements, . . .

Connections between Buchberger's Algorithm and Faugère's F4

Similarities

1. Both compute a Gröbner basis G for a finite set of polynomials F.
2. Both generate pairs of elements of the input, reduce these, add newly generated elements to G, generate new pairs of elements, ...

Differences

1. F4 should select many pairs at a time.
2. F4 pre-computes all reducers for all pairs of the given selection.
3. F4 stores all the above data in a huge matrix M and reduces all pairs simultaneously computing the row echelon form of this matrix.

Pseudocode of basic F4

Require: finite subset F of $\mathbb{K}[\underline{x}]$

$G := F$
$P := \{(g_i, g_j) \mid g_i, g_j \in G, i < j\}$
$d := 1$
while $P \neq \emptyset$ **do**
  $P_d := \mathrm{select}(P)$
  $P := P \setminus P_d$
  $L_d := \{u_i g_i \mid g_i \text{ gen. of pair in } P_d,$
    $u_i \text{ corr. multiple for s-poly}\}$
  $M_d := \mathrm{symbPreprocessing}(L_d, G)$
  $\tilde{M}_d := \mathrm{rowEchelonForm}(M_d)$
  $\tilde{M}_d^+ := \{p \text{ corr. to rows in } \tilde{M}_d \mid \mathrm{lm}(p) \notin L(G)\}$
  $P := P \cup \{(p, g) \mid p \in \tilde{M}_d^+, g \in G\}$
  $G := G \cup \tilde{M}_d^+$
  $d := d + 1$
return Gröbner basis G of F

A first look

PROS

1. Reduction is done for the complete selection at once.
2. Matrix computations can be parallelized.

A first look

PROS

1. Reduction is done for the complete selection at once.
2. Matrix computations can be parallelized.

CONS

1. huge matrices (memory usage!)
2. slow (standard version)

An easy example

Let $F = G = \{g_1, g_2\} \subset \mathbb{K}[x, y, z]$ where

$$g_1 = xy - z^2,$$
$$g_2 = y^2 - z^2.$$

$>$ degree reverse lexicographical ordering $x > y > z$

An easy example

Let $F = G = \{g_1, g_2\} \subset \mathbb{K}[x, y, z]$ where

$$g_1 = xy - z^2,$$
$$g_2 = y^2 - z^2.$$

$>$ degree reverse lexicographical ordering $x > y > z$

Add $g_3 = xz^2 - yz^2$ to G.

Improving F4

Do we need to compute (g_3, g_2) ?

Improving F4

Do we need to compute (g_3, g_2) ?
No, since $\gcd(\mathrm{lm}(g_3), \mathrm{lm}(g_2)) = 1$ (Product Criterion).

Improving F4

Do we need to compute (g_3, g_2) ?
No, since $\gcd(\mathrm{lm}(g_3), \mathrm{lm}(g_2)) = 1$ (Product Criterion).

$\Rightarrow$ Use criteria to discard "redundant" pairs, e.g. Buchberger's criteria

Improving F4

Do we need to compute (g_3, g_2) ?
No, since $\gcd(\mathrm{lm}(g_3), \mathrm{lm}(g_2)) = 1$ (Product Criterion).

$\Rightarrow$ Use criteria to discard "redundant" pairs, e.g. Buchberger's criteria

Problem: We are still too slow!

Pseudocode of full F4

Require: finite subset F of $\mathbb{K}[\underline{x}]$

$G := F$

$P := \text{criteria}(\{(g_i, g_j) \mid g_i, g_j \in G, i < j\})$

$d := 1$

while $P \neq \emptyset$ **do**

$P_d := \text{select}(P)$

$P := P \backslash P_d$

$L_d := \{u_i g_i \mid g_i \text{ gen. of pair in } P_d,$

$u_i \text{ corr. multiple for s-poly}\}$

$M_d := \text{symbPreprocessing}(\text{simplify}(L_d, G))$

$\tilde{M}_d := \text{rowEchelonForm}(M_d)$

$\tilde{M}_d^+ := \{p \text{ corr. to rows in } \tilde{M}_d \mid \text{lm}(p) \notin L(G)\}$

$P := \text{criteria}(P \cup \{(p, g) \mid p \in \tilde{M}_d^+, g \in G\})$

$G := G \cup \tilde{M}_d^+$

$d := d + 1$

return Gröbner basis G of F

Pseudocode of full F4

Require: finite subset F of $\mathbb{K}[\underline{x}]$

$G := F$
$P := \text{criteria}(\{(g_i, g_j) \mid g_i, g_j \in G, i < j\})$
$d := 1$
while $P \neq \emptyset$ **do**
  $P_d := \text{select}(P)$
  $P := P \setminus P_d$
  $L_d := \{u_i g_i \mid g_i \text{ gen. of pair in } P_d,$
    $u_i \text{ corr. multiple for s-poly}\}$
  $M_d := \text{symbPreprocessing}(\text{simplify}(L_d, G))$
  $\tilde{M}_d := \text{rowEchelonForm}(M_d)$
  $\tilde{M}_d^+ := \{p \text{ corr. to rows in } \tilde{M}_d \mid \text{lm}(p) \notin L(G)\}$
  $P := \text{criteria}(P \cup \{(p, g) \mid p \in \tilde{M}_d^+, g \in G\})$
  $G := G \cup \tilde{M}_d^+$
  $d := d + 1$
return Gröbner basis G of F

Idea behind SIMPLIFY

Use already computed data, i.e. use not only $\tilde{M}_d^+$ for the next iteration, but also $\tilde{M}_d$:

Idea behind SIMPLIFY

Use already computed data, i.e. use not only $\tilde{M}_d^+$ for the next iteration, but also $\tilde{M}_d$:

Assume the reducer ug in the symbolic preprocessing at iteration d, where $g \in G$.
If $\exists t$ s.t. $t \mid u$ and $tg \in M_{<d}$

1. $\exists p \in \tilde{M}_{<d}$ representing a (possibly) more reduced version of tg.
2. Rewrite $\frac{u}{t}p$ by ug.
3. Check $\frac{u}{t}p$ for further rewritings.

Problems of SIMPLIFY

1. We need to store all $\tilde{M}_d \Rightarrow$ memory consumption.

Problems of SIMPLIFY

1. We need to store all $\tilde{M}_d \Rightarrow$ memory consumption.
2. It is possible that the newly chosen reducer is not (much) better.

Problems of SIMPLIFY

1. We need to store all $\tilde{M}_d \Rightarrow$ memory consumption.
2. It is possible that the newly chosen reducer is not (much) better.
3. Sometimes other rewritings would be better. Those are possibly hidden by SIMPLIFY.

Problems of SIMPLIFY

1. We need to store all $\tilde{M}_d \Rightarrow$ memory consumption.
2. It is possible that the newly chosen reducer is not (much) better.
3. Sometimes other rewritings would be better. Those are possibly hidden by SIMPLIFY.

$\Rightarrow$ Ideas of SLIMGB:

1. Add some more criteria for the reducer-rewriting, e.g. length of the poly, size of coeffs, etc.

Problems of SIMPLIFY

1. We need to store all $\tilde{M}_d \Rightarrow$ memory consumption.
2. It is possible that the newly chosen reducer is not (much) better.
3. Sometimes other rewritings would be better. Those are possibly hidden by SIMPLIFY.

$\Rightarrow$ Ideas of SLIMGB:

1. Add some more criteria for the reducer-rewriting, e.g. length of the poly, size of coeffs, etc.
2. Store not the whole bunch of data from done computations, but only a list of "good" rewriters.

How to use F5 in F4?

Instead of Buchberger's criteria one can use F5's criteria,

How to use F5 in F4?

Instead of Buchberger's criteria one can use F5's criteria,

+ more "redundant" pairs are detected,

How to use F5 in F4?

Instead of Buchberger's criteria one can use F5's criteria,

+ more "redundant" pairs are detected,
- only in the homogeneous case,

How to use F5 in F4?

Instead of Buchberger's criteria one can use F5's criteria,

+ more "redundant" pairs are detected,
- only in the homogeneous case,
- we have to add some elements g of $\tilde{M}_d \backslash \tilde{M}_d^+$ to G, i.e. $\mathrm{lm}(g) \in L(G)$.

How to use F5 in F4?

Instead of Buchberger's criteria one can use F5's criteria,

+ more "redundant" pairs are detected,
- only in the homogeneous case,
- we have to add some elements g of $\tilde{M}_d \backslash \tilde{M}_d^+$ to G, i.e. $\mathrm{lm}(g) \in L(G)$.

Due to the last point some intermediate matrices can be bigger than in the Buchberger approach $\Rightarrow$ more data needs to be stored.