Implementación de bases de Gröbner no conmutativas en C++ con un poquito de paralelismo

Iván Renison

Facultad de Matemática, Astronomía, Física y Computación
Universidad Nacional de Córdoba

2025-03-06

Polinomios

Dados un alfabeto X y un cuerpo K el conjunto de polinomios se denota $K[X]$.

Algunos polinomios:

$$3 + x - \frac{1}{2}x^2$$

$$5 + 3y - 2xy + x^3y^5$$

Monomios

Monomios

El producto entre las variables es conmutativo.

Monomios

El producto entre las variables es conmutativo.

- $xy = yx$
- $x^3y^5 = xyyxyxyy$
- $casa = asac$

Monomios no conmutativos

Monomios no conmutativos

Se pueden considerar otros monomios en los que el producto entre las variables no conmuta.

- $xy \neq yx$
- $x^3y^5 \neq xyyxyxyy$
- $casa \neq asac$

Polinomios no conmutativos

Son combinaciones lineales con coeficientes en K de monomios no conmutativos

Polinomios no conmutativos

Son combinaciones lineales con coeficientes en K de monomios no conmutativos

Por ejemplo en los racionales:

- $f_0 = x$
- $f_1 = xy + yz$
- $f_2 = 3xyy - 2xzxy + \frac{4}{3}yzzx$

Polinomios no conmutativos

Son combinaciones lineales con coeficientes en K de monomios no conmutativos

Por ejemplo en los racionales:

- $f_0 = x$
- $f_1 = xy + yz$
- $f_2 = 3xyy - 2xzxy + \frac{4}{3}yzzx$

Se pueden sumar:

- $f_0 + f_1 = f_1 + f_0 = x + xy + yz$
- $f_1 + (\frac{1}{2}xy + 2xz) = \frac{3}{2}xy + 2xz + yz$

Polinomios no conmutativos

Son combinaciones lineales con coeficientes en K de monomios no conmutativos

Por ejemplo en los racionales:

- $f_0 = x$
- $f_1 = xy + yz$
- $f_2 = 3xyy - 2xzxy + \frac{4}{3}yzzx$

Se pueden sumar:

- $f_0 + f_1 = f_1 + f_0 = x + xy + yz$
- $f_1 + (\frac{1}{2}xy + 2xz) = \frac{3}{2}xy + 2xz + yz$

Y se pueden multiplicar:

- $f_0 f_1 = xxy + xyz$
- $f_1 f_0 = xyx + yzx$

Polinomios no conmutativos

Son combinaciones lineales con coeficientes en K de monomios no conmutativos

Por ejemplo en los racionales:

- $f_0 = x$
- $f_1 = xy + yz$
- $f_2 = 3xyy - 2xzxy + \frac{4}{3}yzzx$

Se pueden sumar:

- $f_0 + f_1 = f_1 + f_0 = x + xy + yz$
- $f_1 + (\frac{1}{2}xy + 2xz) = \frac{3}{2}xy + 2xz + yz$

Y se pueden multiplicar:

- $f_0 f_1 = xxy + xyz$
- $f_1 f_0 = xyx + yzx$

Se denota con $K\langle X \rangle$ al conjunto de polinomios no conmutativos

Problema principal del trabajo

Dados $G \subseteq K\langle X\rangle$ y $f \in K\langle X\rangle$, determinar si existen $g_1, ..., g_n \in G$ y $c_1, ..., c_n, c'_1, ..., c'_n \in K\langle X\rangle$ tales que

$$f = \sum_{i=1}^{n} c_i g_i c'_i.$$

Problema principal del trabajo

Dados $G \subseteq K\langle X\rangle$ y $f \in K\langle X\rangle$, determinar si existen $g_1, ..., g_n \in G$ y $c_1, ..., c_n, c'_1, ..., c'_n \in K\langle X\rangle$ tales que

$$f = \sum_{i=1}^{n} c_i g_i c'_i.$$

Por ejemplo, si:

- $g_0 = xy + yz$
- $g_1 = yx + zx$
- $G = \{g_0, g_1\}$

Entonces:

- Para $f_4 = xyx - yyx$ si vale

Problema principal del trabajo

Dados $G \subseteq K\langle X\rangle$ y $f \in K\langle X\rangle$, determinar si existen $g_1, ..., g_n \in G$ y $c_1, ..., c_n, c'_1, ..., c'_n \in K\langle X\rangle$ tales que

$$f = \sum_{i=1}^{n} c_i g_i c'_i.$$

Por ejemplo, si:

- $g_0 = xy + yz$
- $g_1 = yx + zx$
- $G = \{g_0, g_1\}$

Entonces:

- Para $f_4 = xyx - yyx$ si vale, porque es igual a $g_0 x - y g_1$.

Problema principal del trabajo

Dados $G \subseteq K\langle X\rangle$ y $f \in K\langle X\rangle$, determinar si existen $g_1, \ldots, g_n \in G$ y $c_1, \ldots, c_n, c'_1, \ldots, c'_n \in K\langle X\rangle$ tales que

$$f = \sum_{i=1}^{n} c_i g_i c'_i.$$

Por ejemplo, si:

- $g_0 = xy + yz$
- $g_1 = yx + zx$
- $G = \{g_0, g_1\}$

Entonces:

- Para $f_4 = xyx - yyx$ si vale, porque es igual a $g_0 x - y g_1$.
- ¿Y para $f_5 = xyz + zyx$ valdrá?

Problema principal del trabajo

Dados $G \subseteq K\langle X\rangle$ y $f \in K\langle X\rangle$, determinar si existen $g_1, ..., g_n \in G$ y $c_1, ..., c_n, c'_1, ..., c'_n \in K\langle X\rangle$ tales que

$$f = \sum_{i=1}^{n} c_i g_i c'_i.$$

Por ejemplo, si:

- $g_0 = xy + yz$
- $g_1 = yx + zx$
- $G = \{g_0, g_1\}$

Entonces:

- Para $f_4 = xyx - yyx$ si vale, porque es igual a $g_0x - yg_1$.
- ¿Y para $f_5 = xyz + zyx$ valdrá? Lo veremos.

¿Qué aplicaciones prácticas tiene?

En identidades sobre un operador asociativo.

Por ejemplo:

Si tenemos un producto asociativo, constantes a, b, c, d y axiomas:

- $aba = a$
- $bab = b$
- $dc = ab$
- $cd = ba$

Nos podemos preguntar si otras igualdades se pueden deducir a partir de estas.

Ideales

Ideales

Sea $G \subseteq K\langle X\rangle$, se define

$$(G) = \left\{ \sum_{i=1}^{n} c_i g_i c_i' : g_1, ..., g_n \in G, c_1, ..., c_n, c_1', ..., c_n' \in K\langle X\rangle \right\}$$

A (G) se lo llama el ideal generado por G.

Problema principal reformulado

Dados $G \subseteq K\langle X\rangle$ y $f \in K\langle X\rangle$, determinar si vale que

$$f \in (G).$$

Problema principal reformulado

Dados $G \subseteq K\langle X\rangle$ y $f \in K\langle X\rangle$, determinar si vale que

$$f \in (G).$$

El problema principal también se llama problema de pertenencia al ideal.

Decidibilidad

Un problema es:

Decidibilidad

Un problema es:

- Decidible si hay un algoritmo que siempre termina y da la respuesta.

Decidibilidad

Un problema es:

- Decidible si hay un algoritmo que siempre termina y da la respuesta.
- Semi-decidible si hay un algoritmo que para los casos afirmativos termina.

Decidibilidad

Un problema es:

- Decidible si hay un algoritmo que siempre termina y da la respuesta.
- Semi-decidible si hay un algoritmo que para los casos afirmativos termina.

Este problema no es decidible en general

Decidibilidad

Un problema es:

- Decidible si hay un algoritmo que siempre termina y da la respuesta.
- Semi-decidible si hay un algoritmo que para los casos afirmativos termina.

Este problema no es decidible en general pero sí es semi-decidible.

Cómo se intenta resolver

- Hay algunos conjuntos G para los cuales hay un algoritmo fácil.

Cómo se intenta resolver

- Hay algunos conjuntos G para los cuales hay un algoritmo fácil. Esos conjuntos se llaman bases de Gröbner.

Cómo se intenta resolver

- Hay algunos conjuntos G para los cuales hay un algoritmo fácil.
  Esos conjuntos se llaman bases de Gröbner.
- Todos los G tienen una base de Gröbner que genera el mismo ideal.

Cómo se intenta resolver

- Hay algunos conjuntos G para los cuales hay un algoritmo fácil.
  Esos conjuntos se llaman bases de Gröbner.
- Todos los G tienen una base de Gröbner que genera el mismo ideal.
- Pero no siempre es finita.

Cómo se intenta resolver

- Hay algunos conjuntos G para los cuales hay un algoritmo fácil.
  Esos conjuntos se llaman bases de Gröbner.
- Todos los G tienen una base de Gröbner que genera el mismo ideal.
- Pero no siempre es finita.
- Cuando es finita se puede calcular.

Cómo se intenta resolver

- Hay algunos conjuntos G para los cuales hay un algoritmo fácil.
  Esos conjuntos se llaman bases de Gröbner.
- Todos los G tienen una base de Gröbner que genera el mismo ideal.
- Pero no siempre es finita.
- Cuando es finita se puede calcular. Y usar el algoritmo fácil
  para resolver el problema.

Algoritmos

Los algoritmos van enumerando la base de Gröbner.

Algoritmos

Los algoritmos van enumerando la base de Gröbner.
Hay dos algoritmos principales:

- Buchberger: va calculando los polinomios de a uno.
  (George M. Bergman, 1978)
- F4: usa álgebra lineal para calcular muchos polinomios a la vez.
  (Xingqiang Xiu, 2012)

Mi librería

Librería `ncgb` de C++.

Mi librería

Librería **`ncgb`** de C++.

- Implementa estructuras para monomios y polinomios no conmutativos con sus operaciones básicas, como **`*`**, **`+`**, **`-`**, **`==`**, **`+=`**, etc.

Mi librería

Librería `ncgb` de C++.

- Implementa estructuras para monomios y polinomios no conmutativos con sus operaciones básicas, como `*`, `+`, `-`, `==`, `+=`, etc.
- Implementa Buchberger y F4.

Mi librería

Librería `ncgb` de C++.

- Implementa estructuras para monomios y polinomios no conmutativos con sus operaciones básicas, como `*`, `+`, `-`, `==`, `+=`, etc.
- Implementa Buchberger y F4.
- Es paramétrica sobre el cuerpo.

Mi librería

Librería `ncgb` de C++.

- Implementa estructuras para monomios y polinomios no conmutativos con sus operaciones básicas, como `*`, `+`, `-`, `==`, `+=`, etc.
- Implementa Buchberger y F4.
- Es paramétrica sobre el cuerpo.
- Para las matrices de un cuerpo arbitrario usa una implementación propia.

Mi librería

Librería `ncgb` de C++.

- Implementa estructuras para monomios y polinomios no conmutativos con sus operaciones básicas, como `*`, `+`, `-`, `==`, `+=`, etc.
- Implementa Buchberger y F4.
- Es paramétrica sobre el cuerpo.
- Para las matrices de un cuerpo arbitrario usa una implementación propia.
- Para los racionales usa la libraría FLINT para las matrices.

Mi librería

Librería `ncgb` de C++.

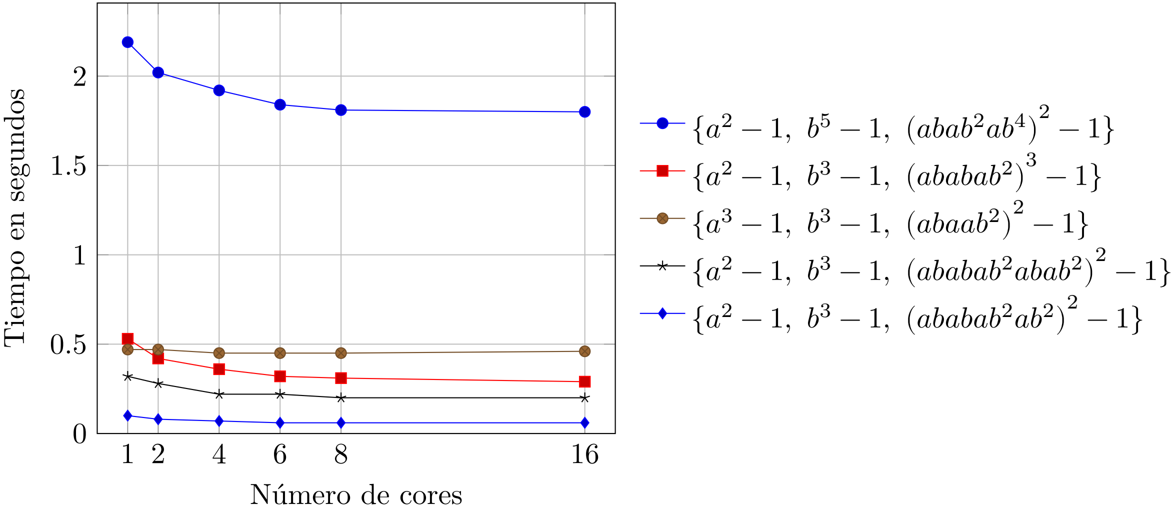
- Implementa estructuras para monomios y polinomios no conmutativos con sus operaciones básicas, como `*`, `+`, `-`, `==`, `+=`, etc.
- Implementa Buchberger y F4.
- Es paramétrica sobre el cuerpo.
- Para las matrices de un cuerpo arbitrario usa una implementación propia.
- Para los racionales usa la libraría FLINT para las matrices.
- Incluye representación de cofactores en Buchberger ("*reconstrucción de la respuesta*").

Mi librería

Librería `ncgb` de C++.

- Implementa estructuras para monomios y polinomios no conmutativos con sus operaciones básicas, como `*`, `+`, `-`, `==`, `+=`, etc.
- Implementa Buchberger y F4.
- Es paramétrica sobre el cuerpo.
- Para las matrices de un cuerpo arbitrario usa una implementación propia.
- Para los racionales usa la libraría FLINT para las matrices.
- Incluye representación de cofactores en Buchberger ("*reconstrucción de la respuesta*").
- Tiene un poquito de paralelismo.

Resultados del paralelismo



Ejemplo de mi librería

Teníamos:

- $g_0 = f_1 = xy + yz$
- $g_1 = f_3 = yx + zx$
- $G = \{g_0, g_1\}$

Dijimos que:

- Para $f_4 = xyx - yyx$ si vale.
- Para $f_5 = xyz + zyx$ no sabíamos si vale o no.

Trabajos futuros

- Implementar la optimización de Faugère-Lachartre en F4.
- Y si es posible paralelizarla.

Fin

Gracias

Implementaciones previas

De Buchberger:

- Paquete GBNP de GAP.
- Implementación dentro de Singular.
- Implementación dentro de NCAlgebra.
- Paquete `OperatorGB` de Mathematica.

De F4:

- Implementación dentro de Magma.
- Paquete `operator_gb` de Python y SageMath.

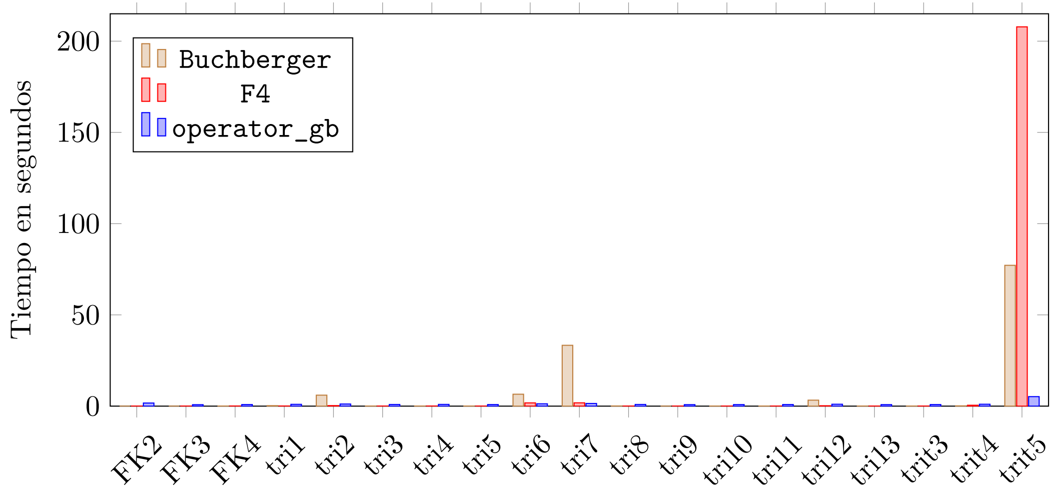
Comparación

Casos de testeo:

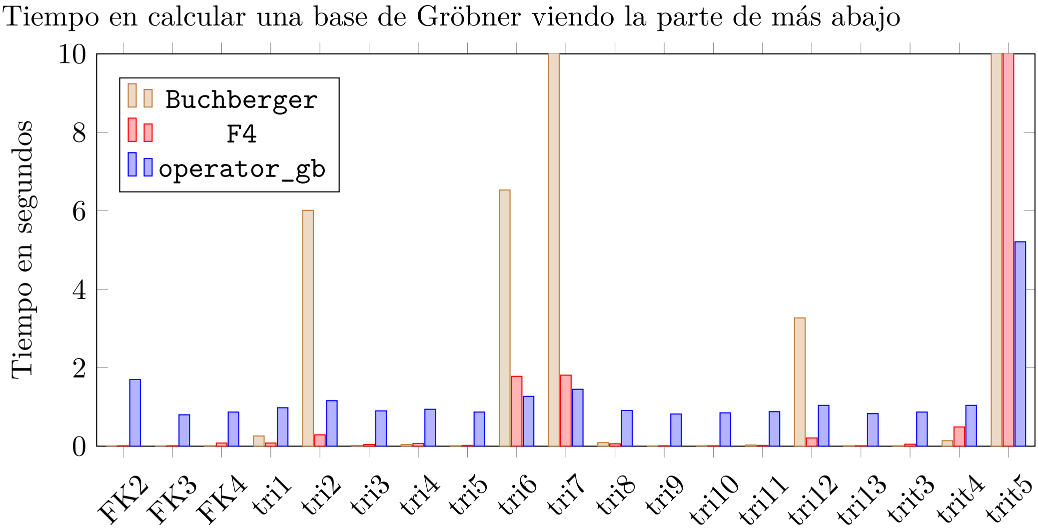
$$
\begin{aligned}
\mathrm{FK2} &= \{a^2\}\\
\mathrm{FK3} &= \{a^2,\ b^2,\ c^2,\ ac+ba+cb,\ ab+bc+ca\}\\
\mathrm{FK4} &= \{a^2,\ b^2,\ c^2,\ d^2,\ e^2,\ f^2,\ ac+ba+cb,\ ae+da+ed,\ bf+db+fd,\ cf+ec+fe,\ ab+bc+ca,\ ad+de+ea,\\
&\qquad bd+df+fb,\ ce+ef+fc,\ cd+dc,\ be+eb,\ af+fa\}\\
\mathrm{tri1} &= \{a^2-1,\ b^3-1,\ (ababab^2ab^2)^2-1\}\\
\mathrm{tri2} &= \{a^2-1,\ b^3-1,\ (ababab^2)^3-1\}\\
\mathrm{tri3} &= \{a^3-1,\ b^3-1,\ (abab^2)^2-1\}\\
\mathrm{tri4} &= \{a^3-1,\ b^3-1,\ (abaab^2)^2-1\}\\
\mathrm{tri5} &= \{a^2-1,\ b^5-1,\ (abab^2)^2-1\}\\
\mathrm{tri6} &= \{a^2-1,\ b^5-1,\ (ababab^4)^2-1\}\\
\mathrm{tri7} &= \{a^2-1,\ b^5-1,\ (abab^2ab^4)^2-1\}\\
\mathrm{tri8} &= \{a^2-1,\ b^2-1,\ (ababab^3)^2-1\}\\
\mathrm{tri9} &= \{a^2-1,\ b^3-1,\ (abab^2)^2-1\}\\
\mathrm{tri10} &= \{a^2-1,\ b^3-1,\ (ababab^2)^2-1\}\\
\mathrm{tri11} &= \{a^2-1,\ b^3-1,\ (abababab^2)^2-1\}\\
\mathrm{tri12} &= \{a^2-1,\ b^3-1,\ (ababab^2abab^2)^2-1\}\\
\mathrm{tri13} &= \{a^2-1,\ b^3-1,\ (babababab^2ab^2)^2-1\}\\
\mathrm{trit3} &= \{a^3-1,\ b^3-1,\ c^3-1,\ (ab)^2-1,\ (ac)^2-1,\ (bc)^2-1\}\\
\mathrm{trit4} &= \{a^3-1,\ b^3-1,\ c^4-1,\ (ab)^2-1,\ (ac)^2-1,\ (bc)^2-1\}\\
\mathrm{trit5} &= \{a^3-1,\ b^3-1,\ c^5-1,\ (ab)^2-1,\ (ac)^2-1,\ (bc)^2-1\}
\end{aligned}
$$

Comparación

Tiempo en calcular una base de Gröbner



Comparación



Comparación

Comparación