Reunión Anual de la Unión Matemática Argentina - Bahía Blanca, 2016

Mercedes Pérez Millán

Introducción a SINGULAR.

Algoritmos y teoría con ejemplos

Aquí presentaremos algunos conceptos del álgebra conmutativa y la geometría algebraica computacional por medio de ejemplos muy sencillos. Para un buen desarrollo de estos temas ver [1, 2, 3].
Heremos referencia constante al manual de SINGULAR [6], que se encuentra en:
`http://www.singular.uni-kl.de/Manual/latest/sing_2336.htm`.

División de polinomios en varias variables y bases de Gröbner

Si tenemos dos polinomios en una variable, x, y queremos dividirlos, primero ordenamos sus términos (monomios), y si tienen grados distintos, sólo podremos dividir al de mayor grado por el de menor grado. Para polinomos en varias variables, pasa algo parecido. Primero necesitamos definir un *orden monomial*. Si tenemos x_1x_2 y x_1x_3, ¿cuál es mayor? hay varios órdenes posibles. El más conocido es el *lexicográfico* con $x_1 > x_2 > x_3 > \cdots > x_n$, donde

$x_1^{\alpha_1}x_2^{\alpha_2}\dots x_n^{\alpha_n} \prec x_1^{\beta_1}x_2^{\beta_2}\dots x_n^{\beta_n}$ si $\alpha_1 < \beta_1$ ó $\alpha_1 = \beta_1$ y $\alpha_2 < \beta_2$ ó ...

Por ejemplo: $x_1x_2 \prec x_1^3x_2$, $x_1x_3 \prec x_1x_2$.

Para ver los órdenes monomiales que encontramos en SINGULAR, buscar en el manual *Term orderings*.

Fijado un orden monomial, que a lo largo de estas notas supondremos es el lexicográfico con $x_1 > x_2 > x_3 > \cdots > x_n$, notamos con $Lt(f)$ al término principal de f (es decir, aquel término cuyo monomio es mayor que todos los monomios de los demás términos de f). Trabajaremos sobre un cuerpo k.

Definición: Sean $\{f_1, \dots, f_s\} \subseteq k[x_1, \dots, x_n]$. Se tiene un *algoritmo de división* con respecto a un orden fijo cuando se tiene un algoritmo tal que dado $f \in k[x_1, \dots, x_n]$ produce $q_1, \dots, q_s, r$ en $k[x_1, \dots, x_n]$ tales que

$$f = q_1f_1 + \cdots + q_sf_s + r$$

con

1. Si $q_i \neq 0, Lt(q_if_i) \preceq Lt(f)$
2. Ningún monomio de r es divisible por $Lt(f_1), \dots, Lt(f_s)$.

Ejemplo:

$$\begin{array}{rlll} & x_1^3x_2 + x_2x_3 & & \big| \, x_1+x_2 \\ - & & & \overline{x_1^2x_2} \\ & \overline{x_1^3x_2} & & \\ & x_2x_3 & & \big| \, x_3-x_4 \\ - & & & \overline{x_2} \\ & x_2x_3 \quad -x_2x_4 & & \\ & \overline{\qquad x_2x_4} & & \\ & \diagup & & \end{array}$$

$$\Rightarrow x_1^3x_2 + x_2x_3 = x_1^2x_2(x_1+x_2) + x_2(x_3-x_4) + x_2x_4$$

donde $Lt(x_1^2x_2(x_1+x_2)) = x_1^3x_2 = Lt(x_1^3x_2+x_2x_3)$, $Lt(x_2(x_3-x_4)) = x_2x_3 \preceq Lt(x_1^3x_2+x_2x_3)$ y $x_1 \nmid x_2x_4$, $x_3 \nmid x_2x_4$.

Observación: Si divido f por $\{f_1, \ldots, f_s\}$ y obtengo $f = q_1f_1 + \cdots + q_sf_s + r$, entonces,

$$f \in \langle f_1, \ldots, f_s \rangle \Leftrightarrow r \in \langle f_1, \ldots, f_s \rangle.$$

Dados $f_1, \ldots, f_s$ polinomios, notemos con $r_{\{f_1,\ldots,f_s\}}(f)$ al resto de dividir a f por $f_1, \ldots, f_s$. Consideremos el ideal I el ideal generado por $f_1, \ldots, f_s$:

$$I = \langle f_1, \ldots, f_s \rangle = \{h_1f_1 + \cdots + h_sf_s : h_i \in k[x_1, \ldots, x_n]\}.$$

Definición: Sea I ideal de $k[x_1, \ldots, x_n]$. Fijado un orden monomial, se dice que $\{g_1, \ldots, g_t\}$ es una *base de Gröbner* de I para dicho orden si

- $g_1, \ldots, g_t \in I$ y,
- vale para cualquier $f \in k[x_1, \ldots, x_n]$ que

$$f \in I \Leftrightarrow r_{\{g_1,\ldots,g_t\}}(f) = 0.$$

(Investigar los comandos `groebner` y `reduce` de SINGULAR.)

Observación: $G = \{g_1, \ldots, g_t\}$ es un sistema de generadores (finito) del ideal I

Dado un orden monomial e $I \subseteq k[x_1, \ldots, x_n]$ ideal, existe el concepto de Base de Gröbner reducida para I, pero no ahondaremos aquí en ese concepto.
(Buscar `redSB` en el manual de SINGULAR.)

Calculemos en SINGULAR la base de Gröbner reducida del ideal $\langle x_1^2 + x_2, x_1 + x_3 - x_4 \rangle$ para el orden lexicográfico con $x_1 > x_2 > x_3 > x_4$.

base reducida ⟶ `option(redSB);`

`ring r=0,(x1,x2,x3,x4),lp;` ⟵ declaramos un anillo de característica 0, con variables $x_1 > x_2 > x_3 > x_4$ y orden lexicográfico

declaramos polinomios ⟶

`poly f1=x1^2+x2;`

`poly f2=x1+x3-x4;`

`ideal J=f1,f2;` ⟵ declaramos el ideal generado por f_1 y f_2

calculamos la base de Gröbner ⟶ `ideal j=groebner(J);`

`j;` ⟵ le pedimos que nos muestre la base

El output es:

```
j[1]=x2+x3^2+x4^2
j[2]=x1+x3+x4
> Auf Wiedersehen.
```

la base de Gröbner es $\{x_2 + x_3^2 + x_4^2, x_1 + x_3 + x_4\}$

Variedades en k^n

Dado un conjunto $V \subseteq k^n$ se dice que es una *variedad* si existe un conjunto de polinomios $\mathcal{F} \subseteq k[x_1, \ldots, x_n]$ tal que $V = \{\mathbf{x} \in k^n : f(\mathbf{x}) = 0 \ \forall f \in \mathcal{F}\}$.

Notemos que $f(\mathbf{x}) = 0$ para todo f en un conjunto de polinomios $\mathcal{F}$ si y sólo si $f(\mathbf{x}) = 0$ para todo f en el ideal $I = \langle \mathcal{F} \rangle$. Dado un conjunto de polinomios $\mathcal{F}$, se define entonces la *variedad del ideal*, $\mathbb{V}(I)$.

Un conjunto finito de puntos es una variedad. En efecto, si $X = \{P_1, \ldots, P_s\}$, con $P_i = (P_{i1}, \ldots, P_{in})$. Sea $I_i = \langle x_1 - P_{i1}, \ldots, x_n - P_{in} \rangle, i = 1, \ldots, s$. Entonces $X = \mathbb{V}(I_1) \cup \cdots \cup \mathbb{V}(I_s) = \mathbb{V}(I_1.I_2.\ldots.I_n)$, donde $\mathbb{V}(I.J) = \{f.h : f \in I, h \in J\}$.

Por otro lado, tenemos lo siguiente:

Definición Sea $X \subseteq k^n$ un conjunto de puntos. Se define el *ideal de* X como

$$\mathbb{I}(X) = \{f \in k[x_1, \ldots, x_n] : f(\mathbf{x}) = 0 \ \forall \mathbf{x} \in X\}$$

Si tenemos un conjunto finito de puntos $X = \{P_1, \ldots, P_s\}$, vale que $\mathbb{I}(X) = I_1 \cap \cdots \cap I_s$, con los ideales $I_i = \langle x_1 - P_{i1}, \ldots, x_n - P_{in} \rangle$ que definimos arriba.

Descomposición primaria de ideales (monomiales libres de cuadrados)

Un ideal $M \subseteq k[x_1, \ldots, x_n]$ se dice *ideal monomial* si está generado por monomios. Un monomio $x_1^{m_1} x_2^{m_2} \ldots x_n^{m_n}$ es *libre de cuadrados* si $m_i \in \{0, 1\}$ para todo i. Un *ideal monomial libre de cuadrados* es un ideal generado por monomios libres de cuadrados.

Definición: Un ideal q se dice *primario* si $f_1 f_2 \in q$ implica que $f_1 \in q$ ó $f_2^m \in q$ para algún $m \in \mathbb{N}$.

Definición: Una *descomposición primaria* de un ideal I es una expresión de I como una intersección finita de ideales primarios.

Propiedades:

- Un ideal q se dice primo si $f_1 f_2 \in q$ implica $f_1 \in q$ o $f_2 \in q$. Si un ideal q es primo, entonces q es primario.
- Si q es de la forma $q = \langle x_{i_1}, \ldots, x_{i_r} \rangle \subset k[x_1, \ldots, x_n]$, entonces q es primo.
- Si M es un ideal monomial libre de cuadrados, sus componentes primarias son ideales primos de la forma $\langle x_{i_1}, \ldots, x_{i_r} \rangle$.

Recordemos el radical de un ideal: $\sqrt{I} = \{f \in k[x_1, \ldots, x_n] : f^m \in I \text{ para algún } m \in \mathbb{N}\}$. Si $M = \bigcap_{i=1}^{r} q_i$ es una descomposición primaria de M, entonces $\sqrt{M} = \bigcap_{i=1}^{r} \sqrt{q_i}$ es la descomposición primaria de $\sqrt{M}$, donde $\sqrt{q_i}$ es primo para todo i. Si M es un ideal monomial

libre de cuadrados, entonces M es radical ($M = \sqrt{M}$) y por lo tanto los ideales primarios de su descomposición primaria son ideales primos.

Para calcular la descomposición primaria de un ideal en SINGULAR es necesario cargar la librería `primdec.lib`. Hay dos algoritmos disponibles para calcular esta descomposición: `primdecGTZ` y `primdecSY` (¡investigar!). Ambos nos devuelven una lista que contiene los ideales primarios q_i y los primos $\sqrt{q_i}$ que, en nuestro caso de ideales M monomiales libres de cuadrados, tenemos $q_i = \sqrt{q_i}$.

Calculemos en SINGULAR la descomposición primaria del ideal radical libre de cuadrados $M = \langle x_1x_2, x_1x_4\rangle$:

Cargamos la librería →
```
LIB "primdec.lib";
ring r=0,(x1,x2,x3,x4),lp;
```
← declaramos un anillo de característica 0, con variables $x_1 > x_2 > x_3 > x_4$ y orden lexicográfico

declaramos polinomios →
```
poly f1=x1*x2;
poly f2=x1*x4;
```
```
ideal M=f1,f2;
```
← declaramos el ideal generado por f_1 y f_2

calculamos la descomposición primaria →
```
list m=primdecGTZ(M);
```
```
m;
```
← le pedimos que nos muestre la descomposición

El output es:

```
[1]:                    ← primer ideal
 [1]:
      _[1]=x1           ← ideal primario: ⟨x1⟩
 [2]:
      _[1]=x1           ← ideal primo: ⟨x1⟩
[2]:                    ← segundo ideal
 [1]:
      _[1]=x4           ↩ ideal primario: ⟨x4, x2⟩
      _[2]=x2           ↪
 [2]:
      _[1]=x4           ↩ ideal primo: ⟨x4, x2⟩
      _[2]=x2           ↪
```

Luego, la descomposición primaria de M es $M = \langle x_1x_2, x_1x_4\rangle = \langle x_1\rangle \cap \langle x_4, x_2\rangle$.

Polinomios en $\mathbb{F}_2[x_1, \ldots, x_n]$

Recordemos el cuerpo de dos elementos, $\mathbb{F}_2 = \{0, 1\}$ con las operaciones

$+$	0	1
0	0	1
1	1	0

$\times$	0	1
0	0	0
1	0	1

Recordemos la *característica* de un cuerpo: es el menor número natural n tal que

$$\overbrace{1 + 1 + \cdots + 1}^{n \text{ veces}} = 0.$$

Si no existe tal n, la característica se define como 0 (como es el caso de $\mathbb{Q}$, $\mathbb{R}$ y $\mathbb{C}$). En el caso de $\mathbb{F}_2$, la característica es 2.

Notemos que cualquier función $f : \mathbb{F}_2^n \to \mathbb{F}$, para $n \in \mathbb{N}$, se puede describir por medio de polinomios (los polinomios interpoladores). Más aún, estos polinomios se pueden tomar con grado menor o igual a 1 en cada variable (porque $x^2 = x$ para todo x, como se puede ver en la segunda tabla).

Veámoslo en un ejemplo. Supongamos que tenemos todas las asignaciones:

$\mathbf{x}$	$f(\mathbf{x})$
$(0,0)$	α_1
$(1,0)$	α_2
$(0,1)$	α_3
$(1,1)$	α_4

Un polinomio en $\mathbb{F}_2[x_1, x_2]$ con grado a lo sumo 1 en cada variable es de la forma:

$$p(x) = a_1 + a_2x_1 + a_3x_2 + a_4x_1x_2.$$

Podemos pensar en un sistema lineal con 4 ecuaciones y 4 incógnitas a_1, a_2, a_3, a_4. O bien pensar armar el polinomio interpolador de la siguiente manera: para cada punto $P = (x_1^*, x_2^*)$, considero el polinomio $\ell = (x_1 + x_1^* + 1)(x_2 + x_2^* + 1)$. Notemos que $\ell(P) = 1$ y $\ell(Q) = 0$ si $Q \neq P$. Así, $p = \alpha_1(x_1 + 1)(x_2 + 1) + \alpha_2x_1(x_2 + 1) + \alpha_3(x_1 + 1)x_2 + \alpha_4x_1x_2$. Para n variables, si $P_i = (x_{1i}^*, \ldots, x_{ni}^*)$ y $\ell_i = \prod_{j=1}^{n}(x_j + x_{ji}^* + 1)$, entonces $p = \sum_{i=1}^{n}\alpha_i\ell_i$.

Ejercicios:

Trabajaremos en $k = \mathbb{F}_2$, el cuerpo de dos elementos.

Ejercicio 1 Sea $M = \langle x_1x_4, x_2x_3, x_3x_4\rangle$, calcular su descomposición primaria.

Ejercicio 2 Se consideran los ideales $i_1 = \langle x_1 + 1, x_2 + 1, x_3, x_4\rangle$, $i_2 = \langle x_1, x_2 + 1, x_3, x_4 + 1\rangle$, $i_3 = \langle x_1 + 1, x_2, x_3 + 1, x_4\rangle$, $i_4 = \langle x_1 + 1, x_2 + 1, x_3 + 1, x_4 + 1\rangle$.

a) Calcular $I = i_1 \cap i_2 \cap i_3 \cap i_4$.

b) Calcular una base de Gröbner de I para el orden monomial lexicográfico, con $x_1 > x_2 > x_3 > x_4$.

c) Sea $f_0 = x_1x_2(1 + x_3)(1 + x_4)$, reducir el polinomio f_0 con respecto a I.

Una aplicación:

Se considera el siguiente modelo (hiper)simplificado de regulación del metabolismo de la lactosa en *E. coli*:

Cuando hay lactosa en el medio donde vive la bacteria, ésta tiende a producir una proteína, la β-galactosidasa, que se encarga de producir glucosa a partir de la lactosa. Cuando en el medio hay glucosa, se produce una proteína represora que impide la producción de β-galactosidasa. (Podemos decir que la bacteria consume la glucosa presente en el medio, sin necesidad de producir β-galactosidasa que obtenga glucosa a partir de la lactosa.)

Hacemos la siguiente correspondencia:

$$
\begin{array}{lcl}
x_1 & \leftrightarrow & \text{glucosa} \\
x_2 & \leftrightarrow & \text{lactosa} \\
x_3 & \leftrightarrow & \beta\text{-galactosidasa} \\
x_4 & \leftrightarrow & \text{proteína represora.}
\end{array}
$$

Nuestras variables podrán tomar dos valores: 0 si no hay (o hay demasiado poco) del componente en cuestión, 1 si hay ("suficiente"). La función que describe la presencia o ausencia de β-galactosidasa es:

$$h(x_1, x_2, x_3, x_4) = x_2(1 + x_4),$$

que depende de x_2 y x_4.

Supongamos que no conocemos esta función y la queremos averiguar. Para esto contamos con algunas mediciones obtenidas experimentalmente. Lo que se mide en el experimento es lo siguiente: la presencia o ausencia de cada uno de los componentes en un cierto tiempo t, y la presencia o ausencia de β-galactosidasa en el tiempo "siguiente" $t + \varepsilon$ (con $\varepsilon > 0$). Contamos con las siguientes mediciones:

x_i a tiempo t	x_3 a tiempo $t + \varepsilon$
$(1, 1, 0, 0)$	1
$(0, 1, 0, 1)$	0
$(1, 0, 1, 0)$	0
$(1, 1, 1, 1)$	0

Una primera pregunta que nos podemos hacer es de qué variables depende h. Llamemos S al *soporte* de h, es decir, al conjunto de las variables de las que depende h. Notemos que, según las mediciones, $h(1, 1, 0, 0) = 1$ y $h(0, 1, 0, 1) = 0$. Como los valores que cambiaron sólo son los de x_1 y x_4 y cambió la imagen, necesariamente h debe depender o bien de x_1, o bien de x_4, o de ambas. Consideremos el monomio x_1x_4. Haciendo el mismo análisis con los datos 1 y 3, y con los datos 1 y 4, obtenemos respectivamente los monomios x_2x_3 y x_3x_4. Armemos el ideal $M = \langle x_1x_4, x_2x_3, x_3x_4 \rangle$. Vale lo siguiente (tarea):

$$M \subseteq \langle S \rangle.$$

Si consideramos la descomposición primaria minimal de M, $M = \bigcap_{i=1}^{r} p_i$, como $M \subseteq \langle S \rangle$ y $\langle S \rangle$ es un ideal primo, para algún i va a valer $p_i \subseteq \langle S \rangle$. Así, del Ejercicio 1 sabemos que

$$\text{o bien } \langle x_1, x_3 \rangle \subseteq \langle S \rangle, \quad \text{o bien } \langle x_2, x_4 \rangle \subseteq \langle S \rangle, \quad \text{o bien } \langle x_3, x_4 \rangle \subseteq \langle S \rangle.$$

Es decir que no sabemos cuál es el verdadero soporte de h, pero lo que sí sabemos es que podemos buscar una función (polinomial, porque estamos trabajando en un cuerpo finito) que interpole los datos de la tabla, y que solo dependa de x_1 y x_3, o que solo dependa de x_2 y de x_4, o que solo dependa de x_3 y de x_4. ¿Las buscamos?

Consideremos primero alguna función polinomial f_0 que interpole. Siguiendo la cuenta que hicimos más arriba, podemos considerar $f_0 = x_1x_2(x_3 + 1)(x_4 + 1)$.

¿Cómo son **todos** los polinomios que interpolan los datos de la tabla? Son de la forma

$$f_0 + g,$$

donde $g \in I$, I el ideal de los puntos de la primera columna de la tabla. Ese ideal se calcula como en el Ejercicio 2 a):

$$I = i_1 \cap i_2 \cap i_3 \cap i_4,$$

con $i_1 = \langle x_1 + 1, x_2 + 1, x_3, x_4 \rangle$, $i_2 = \langle x_1, x_2 + 1, x_3, x_4 + 1 \rangle$, $i_3 = \langle x_1 + 1, x_2, x_3 + 1, x_4 \rangle$, $i_4 = \langle x_1 + 1, x_2 + 1, x_3 + 1, x_4 + 1 \rangle$.

Ahora nos resta elegir un orden monomial apropiado, calcular una base de Gröbner para I con ese orden, y luego buscar el resto de dividir f_0 por esa base. ¡Pero esto es lo que hicimos en los ejercicios 2 b) y c)!

Más ejercicios:

Ejercicio 3 ¿Cuánto se parece el polinomio obtenido en el Ejercicio 2c) al que queríamos obtener ($h = x_2(x_4 + 1)$)? Utilizar otro orden monomial para obtener uno más parecido al que buscamos.

Ejercicio 4 ¿Qué sucede si agregamos más datos? Por ejemplo:

$(1, 0, 0, 0)$	0

Ejercicio 5 ¿Cómo sería la tabla completa de datos? (¡Suponiendo que todos se pueden medir!)

Ejercicio 6 ¿Qué datos conviene agregar al primer dato de la tabla original para obtener rápidamente el soporte deseado?

References

[1] Atiyah M. F., Macdonald I. G. (1969), Introduction to commutative algebra. Series in Mathematics. Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont.

[2] Cox D., Little J., O'Shea D., (1992) Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra, Springer-Verlag, New York.

[3] Cox D. A., Little J., O'Shea D., (2005), Using algebraic geometry. Graduate Texts in Mathematics, -Vol.185,Second Edition,Springer, New York.

[4] Jarrah A. S., Laubenbacher R., Stigler B., Stillman M., (2007), Reverse-engineering of polynomial dynamical systems, Adv. Appl. Math., 39, 477–489.

[5] Laubenbacher R., Stigler B., (2004), A computational algebra approach to the reverse-engineering of gene regulatory networks. Journal of Theoretical Biology, 229, 523–537.

[6] Decker, W.; Greuel, G.-M.; Pfister, G.; Schönemann, H.: SINGULAR 4-0-2 — A computer algebra system for polynomial computations. http://www.singular.uni-kl.de (2015).