



# UNIVERSIDAD DE GRANADA

Facultad de Ciencias

## ENTREGA DE EJERCICIOS 2: BASES ESTÁNDAR

*Doble Grado Ingeniería Informática y Matemáticas*

Autores:

Javier Gómez López  
Pablo Fuentes Jimenez  
Alejandro Rubio Martínez  
Inmaculada García Moreno  
Juan Valentín Guerrero Cano

23 de noviembre de 2023



Este trabajo se distribuye bajo una licencia CC BY-NC-SA 4.0.

Eres libre de distribuir y adaptar el material siempre que reconozcas a los autores originales del documento, no lo utilices para fines comerciales y lo distribuyas bajo la misma licencia.

[creativecommons.org/licenses/by-nc-sa/4.0/](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Sea  $R = \mathbb{C}[x, y]$  con el orden monomial  $>_{dp}$ . Tomamos el polinomio  $f(x, y) = y^5 - x^7 + ax^3y^3 + bx^4y^4$  con  $a, b \in \mathbb{C}$ . Se denomina ideal de Tjurina  $(J)_f$  de  $f$  al ideal

$$\mathcal{J}_f := \langle f, \partial f / \partial x, \partial f / \partial y \rangle$$

- (a) Usando el algoritmo 1, calcula las posibles bases estándar de  $\mathcal{J}_f$  en función del valor de los parámetros  $a$  y  $b$ .

En nuestro caso, hemos desarrollado un algoritmo en **Python**, y posteriormente hemos usado **Singular** para simplificar los resultados obtenidos y comprobarlos.

Nuestro programa de **Python** es el siguiente:

```

1 from sympy import symbols, Poly, lcm, degree, LM, diff, simplify
2 from itertools import combinations
3
4 #Definir las variables
5 x, y, a, b = symbols('x y a b')
6
7 def spoly(f,g,ordering, selected_domain):
8     '''Funcion que devuelve el S-polinomio dados dos polinomios y un orden monomial
9     Args:
10         - f: Primer polinomio
11         - g: Segundo polinomio
12         - ordering: Orden monomial deseado'''
13     poly_f = Poly(f, domain=selected_domain) if f != 0 else 0
14     poly_g = Poly(g, domain=selected_domain) if g != 0 else 0
15
16     lm_f = poly_f.LM(order=ordering).as_expr()
17     lm_g = poly_g.LM(order=ordering).as_expr()
18
19     lcm_result = lcm(lm_f, lm_g)
20
21     spoly = (((lcm_result / (poly_f.LT(order=ordering)[0].as_expr()) * \
22         poly_g.LC(order=ordering)) * poly_f.as_expr()) \
23         - ((lcm_result / (poly_g.LT(order=ordering)[0].as_expr()) * \
24             poly_f.LC(order=ordering)) * poly_g.as_expr()))
25
26     return simplify(spoly)
27
28 def NF_Butcher(f, G, ordering, selected_domain):
29     '''Funcion que devuelve una forma normal usando NFBuchberger'''
30     h = Poly(f, domain=selected_domain) if f != 0 else 0
31     T = G.copy()
32
33
34     while(h != 0):
35         Gh = [g for g in T if lcm(LM(g, order=ordering), \
36             LM(h, order=ordering)) == LM(h, order=ordering)]
37
38         if not Gh:
39             break
40
41         g = Gh[0]
42         Gh.remove(g)
43         h = spoly(h,g, ordering, selected_domain)
44
45     return h
46
47 def get_standart_base(domain, ordering, G, NF_func, spoly_func):
48     '''Función que devuelve una base estándar
49     Args:
50         - selected_domain: Dominio de los coeficientes de nuestro polinomio
51         - ordering: orden monomial deseado
52         - G: nuestro conjunto finito G contenido en domain
53         - NF_func: funcion que nos devuelve una forma normal débil
54         - spoly_func: funcion que nos devuelve el s-polinomio dados dos polinomios'''
55
56     S = G.copy()
57     P = [(f,g) for f,g in combinations(S,2)] #Nuestra lista de pares
58
59     while (len(P) != 0):
60         f,g = P[0]
61         P.remove((f,g))
62         s_polinomio = spoly_func(f,g, ordering, domain)
63
64         h = NF_func(s_polinomio, S, ordering, domain)

```

```

65         if (type(h) != int):
66             h = h.as_expr()
67
68         if h != 0:
69             for f in S:
70                 P.append((h, f))
71             S.append(h)
72     return S
73
74
75
76
77
78 #-----
79 def Tjurina_generator(f, selected_domain):
80     '''Funcion que devuelve el ideal de Tjurina
81     Args:
82         - f: Polinomio sobre el que queremos obtener el generador
83         - selected_domain: Domain de Sympy del polinomio'''
84
85     poly_f = Poly(f, domain=selected_domain) if f != 0 else 0
86
87     result = [] #Lista que devolveremos
88
89     result.append(f)
90
91     diff_x = diff(poly_f, x).as_expr()
92     diff_y = diff(poly_f, y).as_expr()
93
94     result.append(diff_x)
95     result.append(diff_y)
96
97     return result
98 #-----
99 # PROGRAMA PRINCIPAL
100
101
102 selected_domain = 'CC' #Nuestro dominio son los complejos, C[x,y]
103 f = y**5 - x**7 + a*x**3*y**3 + b*x**4*y**4
104 ordering = 'grevlex'
105
106 tjurina = Tjurina_generator(f, selected_domain)
107
108 # Pasamos a obtener nuestra base estándar
109 base = get_standart_base(selected_domain, ordering, tjurina, NF_Butcher, spoly)
110
111 print(base)

```

Vamos a explicar cada función del código:

- **spoly**: Función que calcula el S-polinomio tal y cómo se explica en la referencia del libro.
- **NF\_Butcher**: función que calcula la forma normal de Butcher según la referencia del libro. Será el algoritmo de forma normal que usaremos para obtener nuestra base estándar.
- **get\_standart\_base**: el algoritmo que se encuentra al principio de la hoja implementado en Python.
- **Tjurina\_generator**: función que calcula el ideal de Tjurina dado un polinomio.

Para el estudio de las posibles bases estándar, vamos a distinguir casos:

- $a = b = 0$ .

En este caso, el resultado que nos arroja Python es el siguiente:

$$\{-x^7 + y^5, -7x^6, 5y^4\}$$

Lo que vamos a hacer ahora es usar **Singular** para simplificar nuestra base. Para ello, realizamos los siguientes comandos:

```

1      ring R = (0,a,b),(x,y),dp; //Definimos nuestra localizacion
2      ideal S = -x^7 + y^5, -7*x^6, 5*y^4;
3      S = simplify(S,32); S = simplify(S,2); S;
4

```

Y nos da la siguiente salida:

$$\{-7x^6, 5y^4\}$$

El comando **simplify** simplifica nuestro ideal. El argumento 32 borra los generadores que sus términos líderes son los por los términos líderes de otros generadores. El argumento 2 borra los generadores que sean 0. Para comprobar que el resultado es correcto, usaremos la función **std** de **Singular**, que calcula bases estándar:

```

1      poly f = y^5 - x^7;
2      ideal S = f, jacob(f);
3      std(S);
4

```

Y nos da el siguiente resultado:

$$\{x^6, y^4\}$$

que es el mismo resultado que nuestra base simplificando (salvo constantes). En los casos que siguen, el método que usamos es el mismo.

- $a = 0, b \neq 0$

En este caso, el resultado que nos arroja **Python** es el siguiente:

$$\{bx^4y^4 - x^7 + y^5, 4bx^3y^4 - 7x^6, 4bx^4y^3 + 5y^4, 3x^7 + 4y^5, 13y^5, -195x^3y^4, 91x^6y, -1365x^6, -455x^2y^4\}$$

Una vez simplificado en **Singular**, obtenemos que

$$\{13y^5, -455x^2y^4, -1365x^6, 4bx^4y^3 + 5y^4\}$$

La comprobación de **Singular** nos dice la base estándar correcta es:

$$\{y^5, x^2y^4, x^6, 4bx^4y^3 + 5y^4\}$$

luego, nuestra solución es correcta (salvo constante).

- $a \neq 0, b = 0$

En este caso, el resultado que nos arroja **Python** es el siguiente:

$$\{ax^3y^3 - x^7 + y^5, 3ax^2y^3 - 7x^6, 3ax^3y^2 + 5y^4, -1y^5, y^4(-6ay^3 + 15x^4), -35x^3y^4\}$$

Una vez simplificado en **Singular**, obtenemos que

$$\{-7x^6 + 3ax^2y^3, 3ax^3y^2 + 5y^4, -y^5\}$$

La comprobación de **Singular** nos dice la base estándar correcta es:

$$\{7x^6 - 3ax^2y^3, 3ax^3y^2 + 5y^4, y^5\}$$

luego, nuestra solución es correcta (salvo constante).

- $a \neq 0, b \neq 0$

En este caso, el resultado que nos arroja **Python** es el siguiente:

$$\begin{aligned} & \{ax^3y^3 + bx^4y^4 - x^7 + y^5, 3ax^2y^3 + 4bx^3y^4 - 7x^6, 3ax^3y^2 + 4bx^4y^3 + 5y^4, \\ & ax^3y^3 + 3x^7 + 4y^5, y^3(7ax^3 + 13y^2), y^5(-63a^2x^2y + 336by^4 - 237x^3), y^2(-12a^2x^2y^3 - 8ax^6 + 64by^6 - 60x^3y^2), \\ & y^5(21a + 273bxy), -21a^2x^2y^3 + 49ax^6 + 52by^6, -21a^2x^3y^2 - 35ay^4 + 52bxy^5, y^5(-105ay^3 + 273x^4), \\ & y^6(-230459985a^2y - 1112792499x), y^4(-1857492a^3x^2y + 7930944aby^4 - 7223580ax^3 + 2070432bx^4y), \\ & x^2y^2(-52920ay^3 + 137592x^4), y^4(132300ay^3 - 343980x^4), 147a^2x^2y^5 - 3549by^8, \\ & -1323a^3xy^5 - 91728b^2y^9 + 64701bx^3y^5, xy^5(105019740a^3y + 849493008ax + 4451169996bx^2y), \\ & xy^5(-420078960a^3 + 26368956432bx^2), y^5(26460by^4 + 5292x^3), y^4(2160900by^4 + 432180x^3), \\ & 4096y^6, 2,61310029691576e + 16axy^5, 536870912ay^5, 461569425408x^2y^5, \\ & xy^2(4445280a^2y^3 - 112687848ax^4 - 187813080xy^2), -4226826240y^5, ay^4(8589934592a^2x^2 - 68719476736by^3), \\ & x^2y(9,46987077866619e + 16ay^3 - 2,20963651502211e + 17x^4), \\ & -1770209280x^3y^4, 5310627840ax^2y^3 - 12391464960x^6\} \end{aligned}$$

Una vez simplificado en **Singular**, obtenemos que

$$\{7ax^3y^3 + 13y^5, 49ax^6 + 52by^6 - 21a^2x^2y^3, 536870912ay^5, \\ -112687848ax^5y^2 - 187813080x^2y^4 + 4445280a^2xy^5\}$$

La comprobación de **Singular** nos dice la base estándar correcta es:

$$\{y^5, 49ax^6 + 52by^6 - 21a^2x^2y^3, \\ 180075a^7x^3y^2 + 19307236by^5 + 300125a^6y^4\}$$

En este caso, observamos que no obtenemos la misma solución. Esto se debe a que nuestro programa en **Python** genera una base muy grande (17 elementos) y con unos coeficientes no del todo precisos (los  $10^x$  han perdido decimales). Es por ello que la base que nos arroja tiene 4 generadores, pero puesto que la correcta tiene 3, podemos deducir que la de nuestro programa también lo es pero con falta de precisión.

(b) Representa en  $\mathbb{N}^2$  el ideal líder asociado a cada base estándar.

Para ello cogeremos cada una de las bases obtenidas en cada caso y calcularemos el ideal líder. Es decir, calcular el ideal formado por los monomios líderes de cada elemento de la base estándar y posteriormente representarlos en  $\mathbb{N}^2$ . La representación hará referencia a los exponentes de las variables  $x$  e  $y$  respectivamente. Por ejemplo, en caso de que el monomio líder de un elemento sea  $\{2x^2y^4\}$ , la representación de este monomio en  $\mathbb{N}^2$  sería  $(2,4)$ .

■  $a = 0, b = 0$

En este caso la base que obtuvimos es la siguiente:

$$\{-7x^6, 5y^4\}$$

El ideal líder formado por los monomios líderes de cada elemento sería:

$$\{-7x^6, 5y^4\}$$

Expresado en  $\mathbb{N}^2$  obtendríamos:

$$\{(6, 0), (0, 4)\}$$

■  $a = 0, b \neq 0$

En este caso la base que obtuvimos es la siguiente:

$$\{13y^5, -455x^2y^4, -1365x^6, 4bx^4y^3 + 5y^4\}$$

El ideal líder formado por los monomios líderes de cada elemento sería:

$$\{13y^5, -455x^2y^4, -1365x^6, 4bx^4y^3\}$$

Donde podemos observar, que le único elemento de la base es el que realmente cambia. Expresado en  $\mathbb{N}^2$  obtendríamos:

$$\{(0, 5), (2, 4), (6, 0), (4, 3)\}$$

■  $a \neq 0, b = 0$

En este caso la base que obtuvimos es la siguiente:

$$\{7x^6 - 3ax^2y^3, 3ax^3y^2 + 5y^4, y^5\}$$

El ideal líder formado por los monomios líderes de cada elemento sería:

$$\{7x^6, 3ax^3y^2, y^5\}$$

Expresado en  $\mathbb{N}^2$  obtendríamos:

$$\{(6, 0), (3, 2), (0, 5)\}$$

- $a \neq 0, b \neq 0$

En este caso la base que obtuvimos es la siguiente:

$$\{7ax^3y^3 + 13y^5, 49ax^6 + 52by^6 - 21a^2x^2y^3, 536870912ay^5, \\ -112687848ax^5y^2 - 187813080x^2y^4 + 4445280a^2xy^5\}$$

El ideal líder formado por los monomios líderes de cada elemento sería:

$$\{7ax^3y^3, 49ax^6, 536870912ay^5, -112687848ax^5y^2\}$$

Expresado en  $\mathbb{N}^2$  obtendríamos:

$$\{(3, 3), (6, 0), (0, 5), (5, 2)\}$$

- (c) Haz un resumen de las páginas 121 a la 130 de "A singular introduction to commutative algebra" (Referencia principal de la asignatura)

**Proposición 1** Sea  $M$  un  $A$ -módulo y  $N_1, N_2 \subset M$  submódulos, entonces  $(N_1 + N_2)/N_1 \cong N_2/(N_1 \cap N_2)$ .

**Lema 2** Sea  $M$  un  $A$ -módulo.  $M$  es finitamente generado si y solo si  $M \cong A^n/L$  para un  $n$  adecuado en  $\mathbb{N}$  y un submódulo adecuado  $L \subset A^n$ . Equivalentemente, existe un homomorfismo sobreyectivo  $\phi : A^n \rightarrow M$ .

Sea  $A$  un anillo. Si  $M$  es un  $A$ -módulo finitamente generado, entonces  $M \cong A^n/L$  para algún  $L \subset A^n$ . Si  $L$  es finitamente generado,  $\exists N \subset A^m$  tal que  $L \cong A^m/N$ . Así,  $M$  es isomorfo al cokernel de un homomorfismo  $\phi : A^m \rightarrow A^n$ , con  $\phi$  representado por una matriz  $n \times m$  al fijar bases en  $A^m$  y  $A^n$ .

**Definición 3** Sea  $M$  un  $A$ -módulo. Decimos que  $M$  es de *presentación finita* si existe una matriz  $n \times m$ , denotada por  $\phi$ , tal que  $M$  es isomorfo al cokernel del mapa  $A^m \xrightarrow{\phi} A^n$ . La matriz  $\phi$  se llama *matriz de presentación* de  $M$ . Escribimos  $A^m \xrightarrow{\phi} A^n \rightarrow M \rightarrow 0$  para denotar una presentación de  $M$ .

En Singular, definir una matriz o un módulo puede interpretarse de dos maneras: como la matriz de presentación del módulo factor de  $A^n$  o como el submódulo de  $A^n$  generado por las columnas de la matriz.

**Ejemplo en Singular 4 (submódulos, presentación de un módulo):** Singular diferencia entre módulos y matrices. Un módulo se define por generadores, ya sea con corchetes o como combinación lineal de los generadores canónicos  $\text{gen}(1), \dots, \text{gen}(n)$  de  $A^n$ .

**Código de Ejemplo:**

```
ring A = 0, (x,y,z), dp;
module N = [xy,0,yz],[0,xz,z2]; //submódulo de A^3
show(N); //muestra los generadores como vectores
print(N); //la matriz correspondiente
```

Los módulos pueden sumarse y multiplicarse por un polinomio o un ideal. La adición de módulos implica la suma de módulos, diferente de la suma de matrices.

**Conversión de Tipos:**

```
module M = [xy,yz],[xz,z2]; //submódulo de A^2
matrix MM = M; //conversión automática de tipo
show(N+x*M); //operación de suma de módulos
```

Se pueden convertir matrices a módulos y viceversa con `module(matrix)` y `matrix(module)`.

**Operaciones en Módulos:** Las operaciones en módulos se consideran como operaciones en submódulos. Se puede calcular una presentación de  $M$  como submódulo de  $A^2$  usando `syz(M)`.

**Código de Ejemplo de Presentación:**

```

module K = syz(M); //calcula el kernel de M
show(K); //presentación del módulo M

```

**Lema 5** Sean  $M$  y  $N$  dos  $A$ -módulos con presentaciones

$$\begin{array}{ccccccc} A^m & \xrightarrow{\phi} & A^n & \xrightarrow{\pi} & M & \rightarrow & 0 \\ A^r & \xrightarrow{\psi} & A^s & \xrightarrow{\kappa} & N & \rightarrow & 0. \end{array}$$

- a) Si  $\lambda : M \rightarrow N$  es un homomorfismo de  $A$ -módulos, entonces existen homomorfismos de  $A$ -módulos  $\alpha : A^m \rightarrow A^r$  y  $\beta : A^n \rightarrow A^s$  tales que el siguiente diagrama conmuta:

$$\begin{array}{ccccccc} A^m & \xrightarrow{\phi} & A^n & \xrightarrow{\pi} & M & \rightarrow & 0 \\ \downarrow \alpha & & \downarrow \beta & & \downarrow \lambda & & \\ A^r & \xrightarrow{\psi} & A^s & \xrightarrow{\kappa} & N & \rightarrow & 0 \end{array}$$

Esto es,  $\beta \circ \phi = \psi \circ \alpha$  y  $\lambda \circ \pi = \kappa \circ \beta$ .

- b) Si  $\beta : A^n \rightarrow A^s$  es un homomorfismo de  $A$ -módulos tal que  $\beta(\text{Im}(\phi)) \subset \text{Im}(\psi)$ , entonces existen homomorfismos de  $A$ -módulos  $\alpha : A^m \rightarrow A^r$  y  $\lambda : M \rightarrow N$  tales que el diagrama anterior conmuta.

**Ejemplo SINGULAR 6 (cálculo de Hom)** usando las notaciones del Lema 5.

Obtenemos el siguiente diagrama conmutativo:

$$\begin{array}{ccccccc} \text{Hom}(M, N) & \xrightarrow{\phi^N} & \text{Hom}(A^r, N) & \xrightarrow{\phi_N} & \text{Hom}(A^m, N) & & \\ & & \uparrow \psi^A & & \uparrow \psi_A & & \\ & & \text{Hom}(A^r, A^s) & \xrightarrow{\psi_s} & \text{Hom}(A^m, A^s) & & \\ & & \downarrow j & & \downarrow i & & \\ & & \text{Hom}(A^r, A^t) & \xrightarrow{\psi^t} & \text{Hom}(A^m, A^t) & & \end{array}$$

Los mapas están definidos como en el Lema 2.1.5 (libro 'A Singular Introduction to Commutative Algebra'). En particular,  $\phi_N^*(\sigma) = \sigma \circ \phi$ ,  $\phi^*(\sigma) = \sigma \circ \phi$ ,  $i(\sigma) = \psi \circ \sigma$ , y  $j(\sigma) = \psi \circ \sigma$ . El Lema 5 y la Proposición 2.4.3 (libro 'A Singular Introduction to Commutative Algebra') implican que

$$\text{Hom}(M, N) = \text{Ker}(\phi_N^*) \cong \phi^{*-1}(\text{Im}(i))/\text{Im}(j).$$

Utilizando el comando interno de Singular `modulo`, identificando como antes  $\text{Hom}(A^n, A^s) = A^{sn}$  y  $\text{Hom}(A^m, A^s) = A^{ms}$ , tenemos que  $D := \phi^{*-1}(\text{Im}(i))$  es el núcleo de  $A^{ns} \xrightarrow{\phi^*} A^{ms}/\text{Im}(i)$ , que es dado por una matriz  $ns \times k$  con entradas en  $A$ , y podemos calcular  $\text{Hom}(M, N)$  como

$$\phi^{*-1}(\text{Im}(i))/\text{Im}(j) = \text{Ker}(A^k \xrightarrow{D} A^{ns}/\text{Im}(j)) = A^k \text{ modulo } (D, j).$$

Finalmente, obtenemos el siguiente procedimiento con  $F = \phi^*$ ,  $B = i$ ,  $C = j$ .

```

proc Hom(matrix M, matrix N)
{
    matrix F = kontraHom(M, nrow(N));
    matrix B = kohom(N, ncol(M));
    matrix C = kohom(N, nrow(M));
    matrix D = modulo(F, B);
    matrix E = modulo(D, C);
    return(E);
}

```

**Definición 7** Sea  $M$  un  $A$ -módulo. Entonces se dice que  $M$  es Noetheriano si cada submódulo  $N \subset M$  es finitamente generado.

**Lema 8**



- a) Los submódulos y módulos cociente de módulos Noetherianos son Noetherianos.
- b) Sean  $N \subset M$  módulos de  $A$ , entonces  $M$  es Noetheriano si y solo si  $N$  y  $M/N$  son Noetherianos.
- c) Sea  $M$  un  $A$ -módulo, entonces las siguientes propiedades son equivalentes:
  - (a)  $M$  es Noetheriano.
  - (b) Toda cadena ascendente de submódulos  $M_1 \subset M_2 \subset \dots \subset M_k \subset \dots$  se estaciona, es decir, existe un  $n$  tal que para todo  $k \geq n$ ,  $M_k = M_n$ .
  - (c) Todo conjunto no vacío de submódulos de  $M$  tiene un elemento maximal con respecto a la inclusión.

**Proposición 9** Sea  $A$  un anillo Noetheriano y  $M$  un  $A$ -módulo finitamente generado, entonces  $M$  es un  $A$ -módulo Noetheriano.

**Lema 10 (Nakayama)** Sea  $A$  un anillo e  $I \subset A$  un ideal contenido en el radical de Jacobson de  $A$ . Sea  $M$  un  $A$ -módulo finitamente generado y  $N \subset M$  un submódulo tal que  $M = IM + N$ . Entonces  $M = N$ . En particular, si  $M = IM$  entonces  $M = 0$ .

**Corolario 11** Sea  $(A, m)$  un anillo local y  $M$  un  $A$ -módulo finitamente generado. Sean  $m_1, \dots, m_n \in M$  tales que sus clases forman un sistema de generadores para el espacio vectorial  $A/m$ -vector  $M/mM$ . Entonces  $m_1, \dots, m_n$  generan  $M$ .

**Observación 12** Con las suposiciones del Corolario 11,  $\{m_1, \dots, m_n\}$  es un sistema mínimo de generadores de  $M$  si y solo si sus clases forman una base de  $M/mM$ , y entonces  $n$  es la dimensión del espacio vectorial  $A/m$ -vector  $M/mM$ .

**Definición 13** Sea  $(A, m)$  un anillo local y  $M$  un  $A$ -módulo. Una presentación  $A^m \xrightarrow{\phi} A^n \rightarrow M \rightarrow 0$  de  $M$  se llama una presentación mínima si  $n = \dim_{A/m}(M/mM)$ .

Si queremos minimizar la presentación de un módulo y encontramos un elemento inversible en la matriz de presentación, podemos usar operaciones de fila y columna para cambiar ese elemento a 1 y el resto de su fila y columna a 0. Eliminamos luego esa fila y columna, sin alterar el cokernel del módulo. Repetimos este proceso hasta que todos los elementos inversibles sean eliminados, logrando una presentación mínima. En el caso de que el anillo sea un campo, cualquier elemento no cero puede servir como pivote para una reducción de Gauss completa, lo cual es automatizado por el comando `prune` en Singular.

**Ejemplo 14 de SINGULAR** presentaciones mínimas y el uso del comando `prune`. Se define un anillo local  $A$  con ideal maximal  $\langle x, y, z \rangle$  y se trabaja con un  $A$ -módulo  $M$  generado por vectores con unidades en sus entradas. Al aplicar el comando `prune`, se simplifica la matriz de presentación de  $M$ , eliminando las unidades y reduciéndola a una forma mínima que mantiene la estructura isomórfica del módulo. Esto es análogo a realizar una reducción de Gauss con elementos pivote que son unidades.

```
ring A = 0, (x, y, z), ds; // anillo local con ideal maximal <x, y, z>
module M = [0, xy - 1, xy + 1], [y, xz, xz];
print(M);
// -> 0, y,
// -> -1 + xy, xz,
// -> 1 + xy, xz
print(prune(M));
// -> -y + xy^2,
// -> -2xz
```

Este procedimiento muestra la eficacia de `prune` para obtener una presentación mínima en un anillo local y destaca la diferencia cuando se trabaja en anillos no locales, donde las unidades en la matriz de presentación juegan un rol crucial en la simplificación.

**Corolario 15 (Teorema de la Intersección de Krull)** Sea  $A$  un anillo Noetheriano y  $I \subset A$  un ideal contenido en el radical de Jacobson, y sea  $M$  un  $A$ -módulo finitamente generado. Entonces, la intersección de todos los  $I^k M$  para  $k$  en los números naturales es igual a 0, es decir,

$$\bigcap_{k \in \mathbb{N}} I^k M = 0.$$

Demostración. Sea  $N := \bigcap_k I^k M$ .  $N$  es un  $A$ -módulo finitamente generado, ya que es un submódulo del módulo finitamente generado  $M$  sobre el anillo Noetheriano  $A$ . Por el Lema de Nakayama es suficiente demostrar que  $IN = N$ . Sea

$$\mathcal{M} := \{L \subseteq M \text{ submódulo} \mid L \cap N = N\}$$

Dado que  $A$  es Noetheriano, el conjunto  $\mathcal{M}$  tiene un elemento maximal que llamaremos  $L$ . Queda por demostrar que  $I^k M \subseteq L$  para algún  $k$ , porque esto implica que la cadena se estabiliza ya que  $A$  es Noetheriano.

$$N = I^k M \cap N \subseteq L \cap N = IN$$

. Dado que  $I$  es finitamente generado, basta demostrar que para cualquier  $x \in I$  hay algún entero positivo  $a$  tal que  $x^a M \subseteq L$ . Sea  $x \in I$  y considere la cadena de ideales  $L : I(x) \subseteq L : I(x^2) \subseteq \dots$ . Esta cadena se estabiliza porque  $A$  es Noetheriano.

Elija  $a$  con  $L : I(x^a) = L : I(x^{a+1})$ . Afirmamos que  $x^a M \subseteq L$ . Por la maximalidad de  $L$  es suficiente probar que  $(L + x^a M) \cap N \subseteq IN$  (note que, obviamente,  $IN \subseteq (L + x^a M) \cap N$ ). Sea  $m \in (L + x^a M) \cap N$ , entonces  $m = n + x^a s$ , con  $n \in N$ ,  $s \in M$ . Ahora  $xm - xn = x^{a+1}s \in IN + L = L$ , lo que implica  $s \in L : I(x^{a+1}) = L : I(x^a)$ . Por lo tanto,  $x^a s \in L$  y, consecuentemente,  $m \in L$ . Esto implica  $m \in L \cap N = IN$  ■.

**Definición 16** Sea  $A$  un anillo y  $S \subset A$  un subconjunto multiplicativamente cerrado. Sea  $M$  un  $A$ -módulo.

- Definimos la localización de  $M$  con respecto a  $S$ ,  $S^{-1}M$ , de la siguiente manera:  $S^{-1}M := \{\frac{m}{s} \mid m \in M, s \in S\}$ , donde  $\frac{m}{s}$  denota la clase de equivalencia de  $(m, s) \in M \times S$  con respecto a la siguiente relación de equivalencia:  $(m, s) \sim (m', s')$  si y solo si existe  $t \in S$  tal que  $t(s'm - sm') = 0$ . Además, en  $S^{-1}M$  definimos una adición y multiplicación con elementos del anillo por las mismas fórmulas que para el campo de fracciones (ver antes de la Definición 1.4.4). También usaremos la notación  $M_S$  en lugar de  $S^{-1}M$ . Si  $S = \{1, f, f^2, \dots\}$  entonces escribimos  $M_f$  en lugar de  $S^{-1}M$ . Si  $S = A \setminus P$ ,  $P$  un ideal primo, escribimos  $M_P$  en lugar de  $S^{-1}M$ .
- Sea  $\phi : M \rightarrow N$  un homomorfismo de  $A$ -módulos, entonces definimos el homomorfismo inducido de  $S^{-1}A$ -módulos,  $\phi_S : M_S \rightarrow N_S$ ,  $\frac{m}{s} \mapsto \frac{\phi(m)}{s}$ .

**Proposición 17** Sea  $A$  un anillo y  $S \subset A$  un subconjunto cerrado bajo la multiplicación. Sean  $M, N$   $A$ -módulos y  $\phi : M \rightarrow N$  un homomorfismo de  $A$ -módulos. Entonces:

- El núcleo de  $\phi_S$ , la localización de  $\phi$  con respecto a  $S$ , es igual a la localización del núcleo de  $\phi$  con respecto a  $S$ , es decir,  $\text{Ker}(\phi_S) = \text{Ker}(\phi)_S$ .
- La imagen de  $\phi_S$  es igual a la localización de la imagen de  $\phi$  con respecto a  $S$ , es decir,  $\text{Im}(\phi_S) = \text{Im}(\phi)_S$ .
- El cokernel de  $\phi_S$  es igual al cokernel de  $\phi$  localizado con respecto a  $S$ , es decir,  $\text{Coker}(\phi_S) = \text{Coker}(\phi)_S$ .

En particular, la localización con respecto a  $S$  es un functor exacto. Es decir, si  $0 \rightarrow M' \rightarrow M \rightarrow M'' \rightarrow 0$  es una secuencia exacta de  $A$ -módulos, entonces  $0 \rightarrow M'_S \rightarrow M_S \rightarrow M''_S \rightarrow 0$  es una secuencia exacta de  $A_S$ -módulos.

**Proposición 18** Sea  $A$  un anillo y  $M$  un  $A$ -módulo. Las siguientes condiciones son equivalentes:

- $M = 0$ .
- $M_P = 0$  para todos los ideales primos  $P$ .

c)  $M_m = 0$  para todos los ideales maximales  $m$ .

**Corolario 19** Sea  $A$  un anillo,  $M, N$   $A$ -módulos, y  $\phi : M \rightarrow N$  un homomorfismo de  $A$ -módulos. Entonces  $\phi$  es inyectivo (respectivamente sobreyectivo) si y solo si  $\phi_m$  es inyectivo (respectivamente sobreyectivo) para todos los ideales maximales  $m$ .

**Definición 20** Sea  $A$  un anillo y  $M$  un  $A$ -módulo. El soporte de  $M$ , denotado por  $\text{supp}(M)$ , se define por

$$\text{supp}(M) := \{P \subset A \text{ ideal primo} \mid M_P \neq 0\}$$

.

**Lema 21** Sea  $A$  un anillo y  $M$  un  $A$ -módulo finitamente generado. Entonces

$$\text{supp}(M) = \{P \subset A \text{ ideal primo} \mid P \supset \text{Ann}(M)\} =: V(\text{Ann}(M))$$

.