UNIVERSIDAD
DE GRANADA

Facultad de Ciencas

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

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 en algoritmo en `Python`, y posteriormente hemos usado `Singular` para simplificar los resultados obtenidos y comprobarlos.

Nuestro programa de `Python` es el siguiente:

```python
from sympy import symbols, Poly, lcm, degree, LM, diff, simplify
from itertools import combinations

#Definir las variables
x, y, a, b = symbols('x y a b')

def spoly(f,g,ordering, selected_domain):
    '''Funcion que devuelve el S-polinomio dados dos polinomios y un orden monomial
       Args:
        - f: Primer polinomio
        - g: Segundo polinomio
        - ordering: Orden monomial deseado'''
    poly_f = Poly(f, domain=selected_domain) if f != 0 else 0
    poly_g = Poly(g, domain=selected_domain) if g != 0 else 0

    lm_f = poly_f.LM(order=ordering).as_expr()
    lm_g = poly_g.LM(order=ordering).as_expr()

    lcm_result = lcm(lm_f, lm_g)

    spoly = (((lcm_result / (poly_f.LT(order=ordering)[0].as_expr()) * \
        poly_g.LC(order=ordering)) ) * poly_f.as_expr()) \
            - ((  (lcm_result / (poly_g.LT(order=ordering)[0].as_expr())) \
                * poly_f.LC(order=ordering) ) * poly_g.as_expr())

    return simplify(spoly)

def NF_Butcher(f, G, ordering, selected_domain):
    '''Funcion que devuelve una forma normal usando NFBuchberger'''
    h = Poly(f, domain=selected_domain) if f != 0 else 0
    T = G.copy()


    while(h != 0):
        Gh = [g for g in T if lcm(LM(g, order=ordering), \
            LM(h, order=ordering)) == LM(h, order=ordering)]

        if not Gh:
            break

        g = Gh[0]
        Gh.remove(g)
        h = spoly(h,g, ordering, selected_domain)

    return h

def get_standart_base(domain, ordering, G, NF_func, spoly_func):
    '''Función que devuelve una base estándar
       Args:
       - selected_domain: Dominio de los coeficientes de nuestro polinomio
       - ordering: orden monomial deseado
       - G: nuestro conjunto finito G contenido en domain
       - NF_func: funcion que nos devuelve una forma normal débil
       - spoly_func: funcion que nos devuelve el s-polinomio dados dos polinomios'''

    S = G.copy()
    P = [(f,g) for f,g in combinations(S,2)] #Nuestra lista de pares

    while (len(P) != 0):
        f,g = P[0]
        P.remove((f,g))
        s_polinomio = spoly_func(f,g, ordering, domain)

        h = NF_func(s_polinomio, S, ordering, domain)
```

```
        if(type(h) != int):
            h = h.as_expr()

        if h != 0:
            for f in S:
                P.append((h,f))
            S.append(h)
    return S





#---------------------------------------------------------------
def Tjurina_generator(f, selected_domain):
    '''Funcion que devuelve el ideal de Tjurina
    Args:
        - f: Polinomio sobre el que queremos obtener el generador
        - selected_domain: Domain de Sympy del poliniomio'''

    poly_f = Poly(f, domain=selected_domain) if f != 0 else 0

    result = [] #Lista que devolveremos

    result.append(f)

    diff_x = diff(poly_f,x).as_expr()
    diff_y = diff(poly_f,y).as_expr()

    result.append(diff_x)
    result.append(diff_y)

    return result
#---------------------------------------------------------------
# PROGRAMA PRINCIPAL


selected_domain = 'CC' #Nuestro dominio son los complejos, C[x,y]
f = y**5 - x**7 + a*x**3*y**3 + b*x**4*y**4
ordering = 'grevlex'

tjurina = Tjurina_generator(f, selected_domain)

# Pasamos a obtener nuestra base estándar
base = get_standart_base(selected_domain,ordering,tjurina,NF_Butcher,spoly)

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:

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

```

  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:

```
poly f = y^5 - x^7;
ideal S = f, jacob(f);
std(S);

```

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:

  $$\begin{gathered}\{ax^3y^3 + bx^4y^4 - x^7 + y^5, 3ax^2y^3 + 4bx^3y^4 - 7x^6, 3ax^3y^2 + 4bx^4y^3 + 5y^4, \\ ax^3y^3+3x^7+4y^5, y^3(7ax^3+13y^2), y^5(-63a^2x^2y+336by^4-237x^3), y^2(-12a^2x^2y^3-8ax^6+64by^6-60x^3y^2), \\ y^5(21a + 273bxy), -21a^2x^2y^3 + 49ax^6 + 52by^6, -21a^2x^3y^2 - 35ay^4 + 52bxy^5, y^5(-105ay^3 + 273x^4), \\ y^6(-230459985a^2y - 1112792499x), y^4(-1857492a^3x^2y + 7930944aby^4 - 7223580ax^3 + 2070432bx^4y), \\ x^2y^2(-52920ay^3 + 137592x^4), y^4(132300ay^3 - 343980x^4), 147a^2x^2y^5 - 3549by^8, \\ -1323a^3xy^5 - 91728b^2y^9 + 64701bx^3y^5, xy^5(105019740a^3y + 849493008ax + 4451169996bx^2y), \\ xy^5(-420078960a^3 + 26368956432bx^2), y^5(26460by^4 + 5292x^3), y^4(2160900by^4 + 432180x^3), \\ 4096y^6, 2.61310029691576e + 16axy^5, 536870912ay^5, 461569425408x^2y^5, \\ xy^2(4445280a^2y^3-112687848ax^4-187813080xy^2), -4226826240y^5, ay^4(8589934592a^2x^2-68719476736by^3), \\ x^2y(9.46987077866619e + 16ay^3 - 2.20963651502211e + 17x^4), \\ -1770209280x^3y^4, 5310627840ax^2y^3 - 12391464960x^6\}\end{gathered}$$

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)

Este apartado se adjunta en un PDF a continuación, por facilidad a la hora de la división de trabajo del grupo.