Universidad de Valladolid

FACULTAD DE CIENCIAS

TRABAJO FIN DE GRADO

Grado en Matemáticas

ESTUDIO DEL MÉTODO DE TRIANGULARIZACIÓN DE SISTEMAS DE ECUACIONES DESCRITO POR J.THOMAS

Autor: Francisco Herguedas Hernández
Tutor: José Cano Torres
Año: 2023

Índice

Capitulo 1

1. Introducción

El objetivo de esta memoria es la descripción del llamado algoritmo de Thomas para la descomposición un sistema de ecuaciones algebraicas en subsistemas triangulares simples. El objetivo del algortimo de Thomas es la descripción del espacio de soluciones de un sistema de ecuaciones diferenciales, en esta memoria nos restringiremos al caso de ecuaciones algebraicas. En este caso el algoritmo de Thomas descompone el espacio de soluciones de un sistemas de ecuaciones algebraicas en una únion disjunta de conjuntos de soluciones de sistemas triangulares simples.

Podríamos entender de forma muy grosera que el algoritmo de Thomas es una generalización del método de resolución de sistemas de ecuaciones lineales de Gauss al caso de sistemas de ecuaciones algebraicas no lineales.

Existen en literatura muchos otros métodos de descomposición de sistemas de ecuaciones algebraicos, como pueden ser los métodos de ***Grobner***, o métodos llamado de ***cadenas regulares***. Aunque no entraremos en la comparación entre dichos métodos. Estos últimos métodos están más enfocados a una descomposición del ideal generado por el sistema de ecuaciones con propiedades interesantes que permiten obtener propiedades del conjunto de soluciones, pero no buscan la descomposición disjunta del conjunto de soluciones como es el caso de la descomposición de Thomas.

Cuando tenemos un sistema lineal podemos obtener un sistema equivalente triangular, pero en el caso del algoritmo de Thomas, lo que obtenemos son un conjunto de sistemas triangulares simples de tal forma que el conjunto de soluciones del sistema original es unión disjunta de los conjuntos de soluciones de los sistemas triangulares simples obtenidos. A diferencia del caso lineal, estos sistemas triangulares pueden contener ecuaciones e inecuaciones.

Por tener un contexto, si por ejemplo tenemos un sistema lineal triangular $S = \{E_1, E_2, E_3\}$, siendo,

$$E_1 := 4x_1 + 3x_2 - 3x_3 - 4x_4 = 0,$$

$$E_2 := x_2 + x_3 + 3x_4 = 0,$$

$$E_3 := 5x_3 + 2x_4 = 0.$$

Hemos ordenado las variables como $x_1 > x_2 > x_3 > x_4$.

Para cada x_i denotamos $S_{<x_i}$ el subconjunto de S formado por las ecuaciones de S que involucren únicamente las variables desde x_{i+1} a x_4. La propiedad principal de un sistema lineal triangular es que para describir su conjunto de soluciones, se puede proceder por el método de sustitución inversa, dependiendo de que en el S exista una ecuación con variable máxima x_i entonces, o bien existe un único elemento c_i tal que $(c_i, c_{i+1}, \ldots, c_n)$ es solución de $S_{x_{i-1}}$, o bien para todo c_i, $(c_i, c_{i+1}, \ldots, c_n)$ es solución de $S_{x_{i-1}}$.

En nuestro ejemplo $\forall c_4 \in K$, c_4 es solución de $S_{<x_3}$, pues, $S_{<x_3} = \emptyset$; $\forall c_4 \in Sol(S_{<x_3})$ $\exists! c_3$ tal que $c_3 = \frac{-2c_4}{5}$, es decir sólo hay una única opción para c_3.

En cambio si tenemos un sistema triangular , por ejemplo,

$$E_1 := 4x_1^2 + 3x_2^3 - 3x_3 - 4x_4 = 0,$$

$$E_2 := x_2 + x_3^5 + 3x_4^2 = 0,$$

$$E_3 := (x_4 - 1)x_3^2 + x_4x_3 + 4x_4 = 0.$$

$\forall c_4 \in K$ como $S_{<x_3} = \emptyset$, $c_4 \in Sol(S_{<x_3})$. En este caso cabría esperar que $\forall c_4 \in Sol(S_{<x_3})$ que existan tantos elementos c_3 como el grado en x_3 del polinomio que involucra solamente a las variables x_3 y x_4 de modo que (c_3, c_4) sea solución de $S_{<x_2}$. Pero en cambio esto ocurre únicamente si $c_4 \neq 1$ y $c_4 \neq 0$. Para garantizar esta propiedad, se añade a la condición de ser triangular dos propiedaes más: la de no anulación de los coeficientes principales, y la anulación de los discriminantes.

En comparación con otras descomposiciones, la descomposición de **Thomas** siempre construye una descomposición ***disjunta*** del conjunto de soluciones.

La disyunción de la descomposicion de **Thomas** combinado con las propiedades de los sistemas *simples*, (definición 3.3.1) proporciona una útil forma para encontrar soluciones a los sistemas polinómicos.

A la largo de este trabajo estudiaremos varios temas que tienen siempre como objetivo conseguir un algoritmo completo para la descomposición de Thomas.

En el primer capítulo estudiaremos las resultantes, para empezar sobre una variable y después de más de una variable, esto nos servirá de base para el estudio de subresultantes y también veremos algún resultado que nos servirá de gran ayuda en el capítulo 3 en alguna demostración de algortimos y resultados de gran importancia.

Para finalizar el primer capítulo, llegaremos a una parte mucho más importante, las subresultantes. Las estudiaremos a fondo, empezando por su definición y definiendo notaciones y propiedades para poder ver resultados en más profundidad. Una vez tenemos todas las nociones de subresultantes y propiedades, empezaremos a ver resultados importantes que nos ayudarán a acabar el capítulo, con la parte más importante de todas, donde relaciona

la secuencia polinomial de restos subresultantes(PRS), con las subresultantes. Estos resultados no serán demostrados ya que se escapan de nuestro trabajo y llevaría un trabajo que es demasiado extenso, pero podríamos encontrarlo en el libro Mishra que mencionaremos más tarde. Estos últimos resultados mencionados son de gran importancia para ayudar a demostrar lemas y proposiciones del capítulo 3.

Por último en el capítulo 3, llegamos al tema principal del trabajo, el algoritmo de Thomas.

Para empezar daremos unas definiciones y conceptos previos, donde primeramente dejaremos claro la notación que usaremos hasta el final. También introduciremos la definición de sistemas simples, y daremos ejemplos para saber qué son, una vez conseguido eso veremos la definición de descomposición de Thomas. Para seguir el capítulo veremos la descomposición algebraica de Thomas, dónde veremos el algoritmo Reduce, que es el más importante de todos los que hay y lo veremos en detalle, y nos ayudará a introducir la última parte del trabajo que es la descomposición de Thomas.

Para finalizar veremos todos los subalgortimos que se utilizan en el algoritmo principal, dónde veremos como se programaría y también en un esquema dónde podemos ver todo más claro.

Capitulo 2

2. Resultantes y Subresultantes

2.1. Resultantes

En esta sección veremos las resultantes, primero en una variable ya que nos servirá de base para luego ver la resultante en varias variables. En todo este capítulo, salvo que se diga lo contrario denotaremos por K un dominio de factorización única.

Definición 2.1.1: Un polinomio no nulo $f(x) \in K[x]$ está definido por:

$$f(x) = \sum_{i=1}^{n} a_i x^i,$$

con $a_n \in K$, $a_n \neq 0$ y el grado de f es n.

Lema 2.1.2: Sea K un dominio de factorización única y sean $f, g \in K[x]$,polinomios con grado n y m, mayores que cero respectivamente. Entonces $f(x)$ y $g(x)$ tienen un factor común distinto de una constante si y sólo si existen polinomios al menos uno de ellos distintos de cero, $A, B \in K[x]$ tal que $deg(A) \leq m-1$, $deg(B) \leq n-1$ y cumple $Af + Bg = 0$.

Demostración: Por el lema de Gauss $K[x]$ es un dominio de factorización única.

$\Longrightarrow$) Supongamos que f y g tienen un factor común no constante $h \in K[x]$. Por tanto existen $f_1, g_1 \in K[X]$ tal que, $f = hf_1$ y $g = hg_1$.

Como los grados de f y g son mayores que 0 (estrictamente), los polinomios f_1 y g_1 son no nulos.

Consideramos ahora:

$$g_1 f + (-f_1)g = g_1(hf_1) + (-f_1)(hg_1) = 0.$$

Observamos que $deg(f_1) \leq n-1$ y $deg(g_1) \leq m-1$, debido a que el grado de h es al menos 1, por tanto tomando $A = g_1$ y $B = f_1$ llegamos a lo que buscábamos.

$\Longleftarrow$) Vamos a razonar por reducción al absurdo.

Supongamos que f y g no tienen factores comunes, entonces $mcd(f, g) = 1$. Por tanto existen dos polinomios $R, S \in K[x]$ y un elemento no nulo $c \in K$ tales que,

$$Rf + Sg = c \in K.$$

Por la propiedad de que $A, B \neq 0$, podemos suponer que $B \neq 0$ y por tanto multiplicando por B y usando que $Af + Bg = 0$ (operando $Af = -Bg$), llegamos a que,

$$Bc = B(Rf + Sg) = RBf + SBg = RBf - SAf = (RB - SA)f.$$

De la igualdad $Bc = (RB - SA)f$ implica que $deg(B) \geq deg(f)$, ya que K es un dominio.

Entonces como B es no nulo, y como tenemos que $(RB - SA)f \neq 0$ y sabiendo que $deg(f) = n$, nos indica que el grado de B es $\geq n$, que esto sería absurdo ya que hemos supuesto que es estrictamente menor que n.$\square$

Vamos a definir la siguiente aplicación, donde S_l denota al espacio vectorial de los polinomios de grado a lo sumo l con coeficientes en el cuerpo $Fr(K)$ de fracciones de K.

$$\begin{array}{rccc} \phi\colon S_{m-1} \times S_{n-1} & \longrightarrow & S_{n+m-1} \\ (A, B) & \longmapsto & Af + Bg \end{array} \tag{1}$$

Fijamos las siguientes bases en los espacios vectoriales de los polinomios,

$B(S_{n-1}) = \{x^{n-1}, \ldots, x, 1\}$, $B(S_{m-1}) = \{x^{m-1}, \ldots, x, 1\}$ y $B(S_{m+n-1}) = \{x^{m+n-1}, \ldots, x, 1\}$.

Luego, $B(S_{m-1} \times S_{n-1}) = \{(x^{m-1}, 0), \ldots, (x, 0), (1, 0), \ldots, (0, x^{n-1}), \ldots, (0, x), (0, 1)\}$.

Definición 2.1.3: Dados dos polinomios $f, g \in K[x]$, la matriz de la aplicación ϕ en las bases anteriores se llama matriz de Sylvester. Se denota por $Syl(f, g, x)$ y es la siguiente de tamaño $(n + m) \times (n + m)$:

$$Syl(f, g, x) = \begin{pmatrix} a_n & \cdots & a_0 & \\ \ddots & & \ddots & \\ & a_n & \cdots & a_0 \\ b_m & \cdots & b_0 & \\ \ddots & & \ddots & \\ & b_m & \cdots & b_0 \end{pmatrix}, \tag{2}$$

donde los espacios blancos son cero y se está considerando vectores coordenados como vectores filas.

Definición 2.1.4: Definimos la Resultante de f y g como,

$$det(Syl(f, g, x)) = Res(f, g, x) \in K.$$

Corolario 2.1.5: $Res(f, g, x) = 0$ si y sólo si f y g tienen un factor común.

Demostración:

$\Longrightarrow$) Supongamos que $det(Syl(f,g,x)) = 0$, y esto implica que $Ker(\phi) \neq \{0\}$.

$Ker(\phi) \neq \{0\}$ implica que existen $A', B' \in Fr(K)[x]$ tal que $A'f + B'g = 0$. Multiplicando por un denominador común de los coeficientes de A' y B', obtenemos que existen $A, B \in K[x]$ con $(A,B) \neq (0,0)$ tal que $Af + Bg = 0$ y además $deg(A) < m$ y $deg(B) < n$, y en estas condiciones estamos en virtud de aplicar el lema 2.1.2 y llegamos a que f, g tienen un factor común.

$\Longleftarrow$) Supongamos que f, g tienen un factor común.

Por el lema 2.1.2 existen $(A,B) \in S_{m-1} \times S_{n-1}$, con $(A,B) \neq (0,0)$ tal que $Af + Bg = 0$. Y esto implica que $(A,B) \in Ker(\phi)$, luego $det(\phi) = det(Syl(f,g,x)) = 0$. $\square$

Ejemplo 2.1.6: Veamos como se calcula la Resultante de dos polinomios genéricos de grado dos.

Sean $f(x) = a_0 + a_1x + a_2x^2$ y $g(x) = b_0 + b_1x + b_2x^2$. Ahora calculemos la matriz de Sylvester:

$$Syl(f,g,x) = \begin{pmatrix} a_2 & a_1 & a_0 & 0 \\ 0 & a_2 & a_1 & a_0 \\ b_2 & b_1 & b_0 & 0 \\ 0 & b_2 & b_1 & b_0 \end{pmatrix}, \tag{3}$$

Utilizando la fórmula antes descrita la resultante de f y g es :

$Res(f,g,x) = b_2^2a_0^2 - 2b_2a_0a_2b_0 + a_2^2b_0^2 - b_1b_2a_1a_0 - b_1a_1a_2b_0 + a_2b_1^2a_0 + b_0b_2a_1^2$.

Nota 2.1.7: Vamos a ver casos particulares de las resultantes:

Caso 1) $Res(c_0, g, x) = c_0^m$, si $a_0 \in K \setminus \{0\}$ y $\deg g = m > 0$.

Caso 2) $Res(f, b_0, x) = b_0^n$, si $b_0 \in K \setminus \{0\}$ y $\deg f = n > 0$.

Caso 3) $Res(a_0, b_0, x) = 1$ si, $a_0, b_0 \in K \setminus \{0\}$.

Proposición 2.1.8: Las resultantes $Res(f,g,x)$ y $Res(g,f,x)$ coinciden salvo en el signo, es decir, se tiene que $Res(f,g,x) = (-1)^{nm}Res(g,f,x)$.

Demostración: Es una consecuencia directo del intercambio de filas, de acuerdo con la definición 2.1.4, ya que al principio al mover la primera columna del polinomio g hacemos m cambios, lo mismo pasa con los sucesivos cambios y por eso aparece el nm.$\square$

Corolario 2.1.9: Para $f, g \in K[x]$, con al menos uno de los dos polinomios de grado positivo, existen polinomios $A, B \in K[x]$, tal que $Af + Bg = Res(f,g,x)$.

Demostración: Vamos a distinguir varios casos:

Si $Res(f,g,x) = 0$, es trivial tomando $A = 0$ y $B = 0$, ya que no hay restricciones impuestas sobre estos polinomios.

Si $f = a_0$ y $g = b_0$, es decir, f y g son polinomios constantes, entonces es trivial que existen $A, B \in K[x]$ tal que, $Af + Bg = 1 = Res(a_0, b_0)$.

Si $f = a_0$ y g es un polinomio con grado igual a m utilizando la nota 2.1.7 tenemos que, $Res(f,g) = a_0^m = a_0^{m-1} \cdot a_0 = a_0^{m-1} \cdot f + 0 \cdot g$.

Si f es un polinomio con grado igual a n y $g = b_0$ se resuelve de forma análoga al caso anterior.

Llamemos $S = syl(f,g,x)$. Al multiplicar la matriz S por la matriz adjunta de S traspuesta y obtenemos:

$$(Ad(S))^t \cdot S = det(S) \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}. \tag{4}$$

Denotemos por $H_1, \cdots, H_n$ las filas de la matriz trapuesta de la adjunta de S, se tiene que:

$$H^1 \cdot S = \begin{pmatrix} det(S) \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

es decir,

$$(\, r_0 \cdots r_{m-1} s_0 \cdots s_{n-1}) \cdot S = \begin{pmatrix} det(S) \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Luego, $[B, \phi, B] \cdot [(r(x), s(x))]_{B_{m-1} \times B_{n-1}} = [det(S)]_B$, es decir, $\phi(r(x), s(x)) = det(S)$.

Por tanto tenemos $r(x)f(x) + s(x)g(x) = det(S)$.$\square$

Ahora veremos aplicaciones. La primera tiene que ver con el discriminante. El discriminante nos da información sobre las posibles raíces. Si por ejemplo tenemos una forma cuadrática $f = ax^2 + bx + c \in K[x]$, el discriminante será $D = b^2 - 4ac$. Podemos diferenciar varios casos. En el caso de que D es positivo esto nos indica que hay dos raíces reales distintas. Por otro parte, si $D = 0$ quiere decir que hay una única raíz con multiplicidad dos. Finalmente si D es negativo, no tiene raíces reales, y las dos únicas raíces son complejas y son conjugadas.

Un polinomio es separable si tiene n raíces distintas en el cuerpo.

Lema 2.1.10: Un polinomio $f \in K[x]$, K un dominio de factorización única con característica 0, tiene un factor no constante doble si y sólo si $Resultant(f, f') = 0$.

Demostración: Como K es un dominio de factorización única, entonces,

$f(x) = a \cdot f_1^{e_1}(x) \cdots f_m^{e_m}(x)$, con $a \in K$ e_i números naturales y $f_i(x) \in K[x]$ elementos irreducibles de $K[x]$.

Entonces $Resultant(f, f') = 0$ (Corolario 2.1.5)

$\iff f(x)$ y $f'(x)$ tienen un divisor común de grado positivo

$\iff (\exists 1 \leq i \leq m)$ tal que $f_i(x) | f'(x)$ y $deg(f_i) > 0$

$\iff (\exists 1 \leq i \leq m)$ tal que $f_i(x) | f'(x) = \sum_{k=1}^{m} e_k f_1^{e_1}(x) \cdots f_k^{e_k - 1}(x) \cdots f_m^{e_m}(x)$

$\iff (\exists 1 \leq i \leq m)$ tal que $f_i(x) | e_i f_1^{e_1}(x) \cdots f_i^{e_i - 1}(x) \cdots f_m^{e_m}(x)$

$\iff (\exists 1 \leq i \leq m)$ tal que $e_i \geq 2$

$\iff$ f tiene un factor no constante doble. $\square$

Lema 2.1.11: Un polinomio $f \in K[x]$ es separable si y sólo si su discriminante no es cero.

Sabemos que el discriminante es cero solo si tiene alguna raíz múltiple o de manera equivalente si un polinomio y su derivada son relativamente primos, por tanto es claro que el discriminante y resultante están de alguna forma relacionados.$\square$

Teorema 2.1.12: Dado un polinomio $f = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n \in K[X]$, el discriminante viene dado por:

$$D = \frac{(-1)^{n(n-1)/2}}{a_n} Res(f, f', x),$$

donde f' es la derivada de f.

Veremos ahora un ejemplo genérico de un polinomio de grado 2 calculando el discriminante.

Ejemplo 2.1.13: Sea $f = c + bx + ax^2$. Calcule su discriminante.

Primero calcularemos la derivada de f, es decir, $f' = b + 2ax$. Utilizando la fórmula del teorema 2.1.12 tenemos que,

$$D = \frac{(-1)^{2(2-1)/2}}{a} Res(f, f', x) = \frac{-1}{a} Res(f, f', x),$$

Calcular $Res(f, f', x)$ es calcular el determinante de $\begin{pmatrix} a & b & c \\ 2a & b & 0 \\ 0 & 2a & b \end{pmatrix}$ que al calcular obtenemos $[ab^2 + 4a^2c - (2ab^2)] = 4a^2c - ab^2$.

Luego obtenemos que $D = \frac{-1}{a}(4a^2c - ab^2) = b^2 - 4ac$.

Lema 2.1.14: Un polinomio $f \in K[x]$ tiene un discriminante igual a cero si y sólo si $Res(f, f', x) = 0$, dónde f' es la derivada de f.

2.2. Grado del mcd

En el siguiente capítulo estaremos muy interesadosen poder calcular el grado del máximo común divisor de dos polinomios $f, g \in K[x]$. En esta subsección proponemos un método para calcular este grado quue muestra la relación con la resultante. Sin embargo este método adolece de dos problemas: en la demostración utiliza que K es un cuerpo, y nosotros queremos utilizar el resultado en el caso en que sea dominio de factorización única. El método que proponemos , no sería computonacionalmente efectivo en el caso en que K fuera un anillo de polinomios. Para obtener un método computacionalmente efectivo para el cálculo estudiaremos en las próximas secciones la teoría de las subresultantes.

Teorema 2.2.1: Sea K un cuerpo y sean f y g dos polinomios con $deg(f) = n$ y $deg(g) = m$, y sea $h = mcd(f, g)$ el máximo común divisor de f y g. Luego el $deg(h)$ es el corango de la matriz de sylvester $Syl(f, g, x)$, es decir, la matriz de Sylvester tiene rango, $rango(Syl(f, g, x)) = n + m - deg(h)$.

Demostración: Llamamos L_d al subespacio generado por

$$L\{(1, 0), \ldots, (x^{m-d}, 0), (0, 1), \ldots, (0, x^{n-d})\} \subseteq S_{m-1} \times S_{n-1}.$$

Lema 2.2.2: Se tiene que:

$$d = grado(mcd(f(x), g(x))) \iff \{Ker(\phi) \cap L_d \neq \{0\}\} \wedge \{Ker(\phi) \cap L_{d+j} = \{0\}\}$$

, con $1 \leq j$.

Demostración:

Sea h el mcd de f y g y sea d el grado de h. Como h es el mcd de f y g, entonces existen $f_1, g_1 \in K[x]$ tal que $f(x) = h(x) \cdot f_1(x)$ y $g(x) = h(x) \cdot g_1(x)$ con $deg(f_1) < n$ y $deg(g_1) < m$. Sabemos que $deg(f_1) = n - d$ y $deg(g_1) = m - d$.

Ahora al multiplicar g_1 con f obtenemos, $g_1(x)f(x) = h(x)f_1(x)g_1(x) = f_1(x)g(x)$, es decir, $g_1(x)f(x) - f_1(x)g(x) = 0$.

Por tanto tenemos que $g_1(x)f(x) + (-f_1(x))g(x) = 0 = \phi(g_1(x), (-f_1(x)))$, lo que implica que $(g_1(x), -f_1(x)) \in Ker(\phi) \cap L_d$.

Veamos ahora que si $Ker(\phi) \cap L_i \neq \{0\}$, entonces $i \leq d$.

Supongamos que $Ker(\phi) \cap L_i \neq \{0\}$ por lo tanto, $\exists(\tilde{g}_1(x), -\tilde{f}_1(x)) \neq 0 \in L_d$ tal que $f(x)\tilde{g}_1(x) - g(x)\tilde{f}_1(x) = 0$, con $deg(\tilde{g}_1(x)) \leq m - i$ y $deg(-\tilde{f}_1(x)) \leq n - i$.

También tenemos $f(x) = h(x)f_1(x)$ y $g(x) = h(x)g_1(x)$ con las características anteriores, por tanto llegamos a que $g_1(x)f(x) = h(x)f_1(x)g_1(x) = f_1(x)g(x)$.

Como tenemos varias igualdades que relacionan las distintas funciones que hemos definido tenemos que, $\tilde{g}_1(x)h(x)f_1(x) = \tilde{f}_1(x)h(x)g_1(x)$. Como $h(x) \neq 0$, entonces $\tilde{g}_1(x)f_1(x) = \tilde{f}_1(x)g_1(x)$, es decir, $\tilde{g}_1(x)f_1(x) - \tilde{f}_1(x)g_1(x) = 0$.

Ya que hemos supuesto que $i > d$, tenemos que $deg(\tilde{g}_1(x)) \leq m - i < m - d = grado(g_1(x))$ y $deg(\tilde{f}_1(x)) \leq m - i < m - d = grado(f_1(x))$, lo que quiere decir que $f_1(x), g_1(x)$ tienen un factor común no trivial.

Como $h(x) = m.c.d(f, g)$, entonces $f_1(x), g_1(x)$ son primos entre sí y no tienen factores comunes.

Necesariamente $i \leq d$ si $Ker(\phi) \cap L_i \neq \{0\}$.□

Lema 2.2.3: Si $Ker(\phi) \cap L_d \neq \{0\} \Longrightarrow dim(Ker(\phi)) \geq d$.

Demostración: Siguiendo la demostración de la primera implicación del Lema 2.2.2, tendríamos el sistema:

$$S \cdot \begin{pmatrix} g_{1,0} \\ \vdots \\ g_{1,m-d} \\ 0 \\ \vdots \\ 0 \\ -f_{1,0} \\ \vdots \\ -f_{1,n-d} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Si multiplicaramos el sistema por x bajaríamos cada elemento una fila, es decir, al multiplicar x por la ecuación anterior, obtenemos que $(xg_1(x))f(x) + (-xf_1(x))g(x) = 0$, y en sistema matricial:

$$S \cdot \begin{pmatrix} 0 \\ g_{1,0} \\ \vdots \\ g_{1,m-d} \\ 0 \\ \vdots \\ 0 \\ -f_{1,0} \\ \vdots \\ -f_{1,n-d} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Siguiendo este procedimiento, haciendolo d veces obtenemos d vectores linealmente independientes (ya que son escalonados) que pertenecen al núcleo de S. Por tanto $dim(Ker(S)) \geq d$.$\square$

Sabemos que $Ker(\phi) \cap L_d \neq \{0\}$ y $Ker(\phi) \cap L_{d+1} = \{0\}$ implica que $dim(Ker(\phi)) \geq d$, por lo cual llegamos a que $Rango(S) \geq m + n - d$. $\square$

Vamos a trabajar ahora sobre la resultante multivariable. En la primera parte del capítulo hemos trabajado sobre la resultante en una variable, es decir, polinomios en $K[x]$, dónde toman dos polinomios sobre la variable x y nos devuelve un polinomio sin variables. En el caso de las multivariables, se elimina una incógnita. Veremos primero un ejemplo:

Ejemplo 2.2.4: Calcula la resultante respecto de x de los dos polinomios siguientes: $f(x,y) = x^2y + xy + 3x + 2, g(x,y) = -xy^2 + xy - 1 \in K[x,y]$.

Para calcular $Resultant(f,g,x)$ tenemos que calcular el determinante de $\begin{pmatrix} 2 & -1 & 0 \\ 3+y & y-y^2 & -1 \\ y & 0 & y-y^2 \end{pmatrix}$ que al calcular obtenemos $2y^4 - 5y^3 + 4y$.

La resultante de dos polinomios con dos variables es un polinomio de una variable, ya que la variable respecto la cual tomamos la resultante se elimina.

Ejemplo 2.2.5: Sean $f(x,y) = x^2y^2 - 25x^2 + 9$, $g(x,y) = 4x + y$ dos polinomios en $K[x,y]$. Calcularemos las resultantes para encontrar una raíz común de $f(x,y)$ y $g(x,y)$.

$$Resultant(f,g,x) = \begin{vmatrix} y^2-25 & 0 & 9 \\ 4 & y & 0 \\ 0 & 4 & y \end{vmatrix} = y^4 - 25y^2 + 144$$

$$Resultant(f,g,y) = \begin{vmatrix} x^2 & 0 & -25x^2+9 \\ 1 & 4x & 0 \\ 0 & 1 & 4x \end{vmatrix} = 16x^4 - 25x^2 + 9$$

Las cuatro raíces de $Resultant(f,g,x)$ son $y = \pm 3, \pm 4$ y las de $Resultant(f,g,y)$ son $x = \pm\frac{3}{4}, \pm 1$. Mirando las posibles 16 combinaciones de soluciones, vemos que las posibles soluciones al sistema homogéneo

$$\begin{aligned} f(x,y) &= 0 \\ g(x,y) &= 0 \end{aligned}$$

son $(x,y) = (1,-4), (-1,4), (\frac{3}{4},-3), (-\frac{3}{4},3)$.

Definición 2.2.6: Sean K y K' dos anillos conmutativo y sea $\phi : K \to K'$ un homomorfismo de anillos. Entonces ϕ induce un homomorfismo entre $K[x]$ y $K'[x]$:

$$\begin{aligned} a &\mapsto \phi(a), \\ a_m x^m + \cdots + a_0 &\mapsto \phi(a_m)x^m + \cdots + \phi(a_0), \end{aligned}$$

donde $a, a_m, \cdots, a_0 \in K$.

Lema 2.2.7: Sea $A(x) = a_m x^m + \cdots + a_0$ y $B(x) = b_n x^n + \cdots + b_0$ dos polinomios en el anillo K con grados positivos m y n respectivamente. Si

$$deg(\phi(A)) = m \text{ y } deg(\phi(B)) = k, \ (0 \leq k < n),$$

Entonces,

$$\phi(Resultant(A, B)) = \phi(a_m)^{n-k} \cdot Resultant(\phi(A), \phi(B)).$$

Observación: El grado de A no disminuye.

Demostración: Sea:

$$A' = \phi(A) = \phi(a_m)x^m + \phi(a_{m-1})x^{m-1} + \cdots + \phi(a_0), \text{ y}$$
$$B' = \phi(B) = \phi(b_n)x^n + \phi(b_{n-1})x^{n-1} + \cdots + \phi(b_0).$$

La matriz de Sylvester de $A(x)$ y de $B(x)$ es:

$$M = \begin{pmatrix} a_m & \cdots & a_0 & \\ \ddots & & \ddots & \\ & a_m & \cdots & a_0 \\ b_n & \cdots & b_0 & \\ \ddots & & \ddots & \\ & b_n & \cdots & b_0 \end{pmatrix}, \tag{5}$$

Que tiene $n + m$ filas

y M' la matriz de Sylvester de $A'(x)$ y de $B'(x)$,

$$M' = \begin{pmatrix} \phi(a_m) & \cdots & \phi(a_0) & \\ \ddots & & \ddots & \\ & \phi(a_m) & \cdots & \phi(a_0) \\ \phi(b_k) & \cdots & \phi(b_0) & \\ \ddots & & \ddots & \\ & \phi(b_k) & \cdots & \phi(b_0) \end{pmatrix}, \tag{6}$$

Que tiene $k + m$ filas.

La matriz M' es obtenida de M siguiendo el proceso:

a) La matriz $\phi(M)$ es calculada al sustituir a_i por $\phi(a_i)$, para todo $0 \leq i \leq m$ y reemplazando b_j por $\phi(b_j)$, para todo $0 \leq j \leq n$. Por el enunciado tenemos que:

$$\phi(b_n) = \cdots = \phi(b_{k+1}) = 0.$$

b) De $\phi(M)$ las primeras $(n - k)$ filas y $(n - k)$ columnas son eliminadas, construyendo una matriz $(m + k) \times (m + k)$ igual a M'. Por tanto:

$$\phi(M) =$$

$$= \begin{pmatrix} \phi(a_m) & \cdots & \phi(a_1) & \phi(a_0) & & \\ & \ddots & \ddots & \ddots & & \\ & & \phi(a_m) & \cdots & \phi(a_0) & \\ & & & \phi(a_m) & \cdots & \phi(a_0) \\ 0 & & & M' & & \\ & & & & & \\ & & & & & \\ & & & & & \end{pmatrix}$$

Por lo tanto,

$$\begin{aligned} \phi(Resultant(A,B)) &= \phi(det(Sylvester(A,B))) \\ &= det(\phi(M)) = \phi(a_m)^{n-k} \cdot det(\phi(M')) \\ &= \phi(a_m)^{n-k} \cdot \phi(det(Sylvester(A',B'))) \\ &= \phi(a_m)^{n-k}\phi(Resultant(A',B')). \ \square \end{aligned}$$

2.3. Determinante polinomial

Definición 2.3.1: Sea $M \in K^{m\times n}$ una matriz $m \times n$ con elementos en K. Definimos $M^{(i)} \in K^{m\times m}$, para $i = m, \cdots n$, como la matriz cuadrada $m \times m$ de M que consiste en las primeras $(m-1)$ columnas de M y la i^{th} columna de M, es decir,

$$M^i = \begin{pmatrix} M_{1,1} & \cdots & M_{1,(m-1)} & M_{1,i} \\ M_{2,1} & \cdots & M_{2,(m-1)} & M_{2,i} \\ \vdots & \ddots & \vdots & \vdots \\ M_{m,1} & \cdots & M_{m,(m-1)} & M_{m,i} \end{pmatrix}$$

El determinante polinomial de M es,

$$DetPol(M) = \sum_{i=m}^{n} det(M^{(i)})x^{n-1}.$$

Vemos que $DetPol(M) = 0$, si $n < m$, ya que aparecen dos columnas iguales en los coeficientes de $DetPol(M)$. Se cumple que $deg(DetPol(M)) \leq n - m$, y se cumple la igualdad cuando $det(M^{(m)}) \neq 0$.

Sean $A_1(x), \cdots, A_m(x)$ un conjunto de polinomios en $K[x]$ y sea n el máximo de los grados de los polinomios más uno, es decir,

$$n = 1 + max_{1 \leq i \leq m} deg(A_i)$$

La matriz de $A_1, \cdots, A_m$, $M = Matrix(A_1, \cdots, A_m) \in S^{mxn}$ está definida por:

$$M_{ij} = \text{coeficiente de } x^{n-j} \text{ en } A_i(x).$$

Definimos el determinantes polinomial de $DetPol(A_1, \cdots, A_m) = DetPol(Matrix)(A_1, \cdots, A_m))$.

El determinante polinomial de dos polinomios $A_1, \cdots, A_m$ viene dado por:

$$DetPol(A_1, \cdots, A_m) = DetPol(Matrix(A_1, \cdots, A_m)).$$

$Matrix(A_1, A_2)$ de dos polinomios A_1 y A_2 vienen definido como en el ejemplo 2.3.3.

Propiedades 2.3.2: El determinante polinomial cumple las siguientes propiedades:

1) Para un polinomio $A(x) = a_m x^m + \cdots + a_0 \in K[x]$:

$$Matrix(A) = [a_m, \cdots, a_0], \text{ una } 1 \times (m+1) \text{ matrix;}$$

$$DetPol([a_m, \cdots, a_0]) = a_m x^m + \cdots + a_0 = A(x).$$

2) $DetPol(\cdots, A_i, \cdots, A_j, \cdots) = -DetPol(\cdots, A_j, \cdots, A_i, \cdots)$.

3) Para $a \in K$,

$$DetPol(\cdots, a \cdot A_i, \cdots) = a \cdot DetPol(\cdots, A_i, \cdots).$$

4) Para $a_1, \cdots, a_{i-1}, a_{i+1}, \cdots, a_m \in K[x]$,

$$DetPol(\cdots, A_{i-1}, A_i + \sum_{j=1, i \neq j}^{m} a_j A_j, A_{i+1}, \cdots) = DetPol(\cdots, A_{i-1}, A_i, A_{i+1}, \cdots).$$

Ejemplo 2.3.3: Sean $A_1(x) = 5x^5 + 4x^4 + 3x^3 + 2x^2 + x + 1$ y $A_2(x) = x^2 + x + 1$, entonces:

$$Matrix(A_1, A_2) = \begin{pmatrix} 5 & 4 & 3 & 2 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Teorema 2.3.4: Sean $A(x)$ y $B(x) \neq 0$ dos polinomios en $K[x]$ con grados k y n respectivamente. Sea m un entero que es al menos k y sea

$$\delta = max(m - n + 1, 0) \text{ y } \delta' = max(k - n + 1, 0).$$

Entonces,

$DetPol(x^{m-n}B, x^{m-n-1}B, \cdots, B, A) = b_n^{\delta-\delta'} DetPol(x^{k-n}B, x^{k-n-1}B, \cdots, B, A)$.

Demostración: Esta demostración se divide en tres casos, en el que sólo demostraremos el caso dos,las otras dos demostraciones son análogas.

1. $k < n$.

2. $k \geq n = 0$.

$$\begin{aligned} DetPol(x^{m-n}B, \cdots, B, A) &= 0 \\ &= DetPol(x^{k-n}B, \cdots, B, A) \\ &= b_n^{\delta-\delta'} DetPol(x^{k-n}B, \cdots, B, A). \end{aligned}$$

3. $k \geq n > 0$. □

En el siguiente resultado se comienza a describir la relación entre la resultante y el proceso de pseudo-división.

Teorema 2.3.5: Sean $A(x)$ y $B(x) \neq 0$ dos polinomios con respectivos grados positivos m y n y $b_n = init(B)$. Sea $\delta = max(m - n + 1, 0)$. Luego el pseudo resto de $A(x)$ y $B(x)$ está dado por:

$$b_n^{\delta} PRemainder(A, B) = b_n^{\delta} DetPol(x^{m-n}B, \cdots, B, A).$$

Demostración: Sea,

$$PQuotient = Q(x) = q_{m-n}x^{m-n} + \cdots + q_0$$

$$PRemainder(A, B) = R(x).$$

Entonces,

$$\begin{aligned} b_n^{\delta} \cdot A(x) &= (q_{m-n}x^{m-n} + \cdots + q_0) \cdot B(x) + R(x) \\ &= q_{m-n}x^{m-n}B(x) + \cdots + q_0 B(x) + R(x). \end{aligned}$$

Podemos ver entonces,

$$\begin{aligned} &b_n^{\delta} DetPol(x^{m-n}B, \cdots, B, A) \\ &= DetPol(x^{m-n}B, \cdots, B, b_n^{\delta}A) \text{ (Propiedades 2.3.2)} \end{aligned}$$

$$= DetPol(x^{m-n}B, \cdots, B, b_n^{\delta}A - q_{m-n}x^{m-n}B - \cdots - q_0 B) \text{ (Propiedades)}$$
$$= DetPol(x^{m-n}B, \cdots, B, R) \text{ (Sustituyendo)}$$
$$= b_n^{\delta}R.$$

La última igualdad se justifica con la siguiente relación:

$Matrix(x^{m-n}B, \cdots, B, R) =$

$$= \begin{pmatrix} b_n & b_{n-1} & \cdots & b_0 & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & b_n & b_{n-1} & \cdots & b_0 \\ & & & & & \\ & & & r_p & \cdots & r_0 \end{pmatrix}.$$

Donde,

$$B(x) = b_n x^n + b_{n-1}x^{n-1} + \cdots + b_0$$
$$R(x) = r_p x^p + r_{p-1}x^{p-1} + \cdots + r_0 \text{ y } p < n.$$

La última igualdad de las relaciones puede parecer un poco díficil de entender, por lo tanto, veremos un caso práctico para verlo mejor:

Si A es un polinomio de grado 3, B es un polinomio de grado 2 y R es un polinomio de grado 1, es decir, $A(x) = a_3x^3 + a_2x^2 + a_1x + a_0$, $B(x) = b_2x^2 + b_1x + b_0$ y $R(x) = r_1x + r_0$, por tanto tendríamos:

$$Matrix(xB, B, R) = \begin{pmatrix} b_2 & b_1 & b_0 & 0 \\ 0 & b_2 & b_1 & b_0 \\ 0 & 0 & r_1 & r_0 \end{pmatrix}.$$

Calculando el Detpol sería: $Det(M^3)x^{4-3} + det(M^4)x^{4-4} = r_1b_2^2x + r_0b_2^2 = Rb_2^2$, que es lo que buscábamos.□

Corolario 2.3.6: En el anterior teorema si b_n no es divisor de cero, podemos cancelar b_n^{δ} en ambas partes de la igualdad y obtener:

$$PRemainder(A, B) = DetPol(x^{m-n}B, \cdots, B, A).$$

Teorema 2.3.7: Sean $A(x)$ y $B(x) \neq 0$ dos polinomios en $K[x]$ de grados m y n respectivamente y $b_n = init(B)$. Sean:

$$\delta = max(m - n + 1, 0),$$
$$k = deg(\phi(A)) \leq m,$$
$$n = deg(\phi(B)), \text{ y}$$
$$\delta' = max(k - n + 1, 0).$$

Entonces,

$$\phi(b_n)^{\delta}\phi(PRemainder(A, B)) = \phi(b_n)^{2\delta-\delta'}PRemainder(\phi(A), \phi(B)).$$

El homomorfismo utilizado es el de la definición 2.2.6.

Demostración:

$$\begin{aligned}
&\phi(b_n)^{\delta}\phi(PRemainder(A, B)) \\
&\quad = \phi(b_n^{\delta}PRemainder(A, B)) \\
&\quad = \phi(b_n^{\delta}DetPol(x^{m-n}B, \cdots, B, A)) \text{ (Teorema 2.3.5)} \\
&\quad = \phi(b_n)^{\delta}DetPol(x^{m-n}\phi(B), \cdots, \phi(B), \phi(A)) \\
&\quad = \phi(b_n)^{2\delta-\delta'}DetPol(x^{k-n}\phi(B), \cdots, \phi(B), \phi(A)) \text{ (Teorema 2.3.4)} \\
&\quad = \phi(b_n)^{2\delta-\delta'}PRemainder(\phi(A), \phi(B)). \text{ (Teorema 2.3.5) } \square
\end{aligned}$$

Corolario 2.3.8: En las condiciones del anterior teorema, si $\phi(b_n)$ no es un divisor de cero en K', entonces

$$\phi(PRemainder(A, B)) = \phi(b_n)^{\delta-\delta'}PRemainder(\phi(A), \phi(B))$$

Corolario 2.3.9: Si en las condiciones del anterior teorema $\phi(b_n)$ no es un divisor de cero de K', entonces:

$$\phi(PRemainder(A, B)) = \phi(b_n)^{\delta-\delta'}PRemainder(\phi(A), \phi(B)).$$

2.4. Sucesión de restos polinomiales

Definición 2.4.1: Dos polinomios $P(x), Q(x) \in K[x]$ son similares si existen $A, B \in K$ no nulos tales que, $AP(x) = BQ(x)$, y se denota por $P(x) \sim Q(x)$. Si A y B son unidades de K, entonces los polinomios son asociados.

Definición 2.4.2: Dados dos polinomios $F_1, F_2 \in K[x]$, con $deg(F_1) \geq deg(F_2)$, la sucesión $F_1, \cdots, F_n$ de los polinomios no nulos es la sucesión de pseudo restos de F_1 y F_2 si tenemos lo siguiente:

1) Para todo $i = 3, \cdots, n$,

$$F_i \sim PRemainder(F_{i-2}, F_{i-1}) \neq 0.$$

2) La secuencia termina cuando $PRemainder(F_{n-1}, F_n) = 0$.

Vamos a ver dos procedimientos que se utilizan para obtener esta secuencia, aunque ninguno de los dos son óptimos ya sea porque son costosos por número de operaciones o porque se hacen muy grandes sin utilizar aproximaciones:

Sucesión de los pseudo restos de Euclídes (EPRS): Viene dada por:

$F_i = PRemainder(F_{i-2}, F_{i-1}) \neq 0\ i = 3, \cdots, n$ y $PRemainder(F_{n-1}, F_n) = 0$, el problema es que los coeficientes se van haciendo cada vez más grandes.

Ejemplo 2.4.3: Si empezamos con los polinomios $F_1 = x^8 + x^6 - 3x^4 - 3x^3 + 8x^2 + 2x - 5$ y $F_2 = 3X^6 + 5X^4 - 4X^2 - 9X + 21.$, la sucesión de pseudo restos de Euclídes nos da:

$$F_3 = -15x^4 + 3x^2 - 9,$$

$$F_4 = 15795x^2 + 30375x - 59535,$$

$$F_5 = 1254542875143750x - 1654608338437500,$$

$$F_6 = 12593338795500743100931141992187500.$$

Sucesión primitiva de los pseudo restos PPRS: Viene dada por lo siguiente:

$F_i = Primitive(PRemainder(F_{i-2}, F_{i-1})) \neq 0\ i = 3, \cdots, n$ y $PRemainder(F_{n-1}, F_n) = 0$

El problema desde el punto de vista computacional de esta sucesión es el cálculo en cada paso de mcd, es decir, si K es un anillo de polinomios en varias variables puede ser muy costoso.

Hablaremos de la siguiente sucesión de pseudo restos que es con la que trabajaremos más en profundidad.

Definición 2.4.4: (Sucesión de los pseudo restos subresultantes PRS) Sea K un dominio de factorización única, $F_1, F_2 \in K[x]$ dos polinomios no nulos con $deg(F_1) \geq deg(F_2)$. Sean $F_1, F_2, \cdots, F_n$ definidas por las siguientes condiciones iniciales:

$$\triangle_1 = 0, \triangle_2 = deg(F_1) - deg(F_2) + 1,$$

$$b_1 = 1, b_2 = ld(F_2)$$

$$\psi_1 = 1, \psi_2 = (b_2)^{\triangle_2 - 1}$$

y,

$$\beta_1 = \beta_2 = 1, \beta_3 = (-1)^{\triangle_2},$$

y las siguientes recurrencias,

-Para $i = 3, \cdots, n$,

$$F_i = \frac{PRemainder(F_{i-2}, F_{i-1})}{\beta_i}$$

$$\triangle_i = deg(F_{i-1}) - deg(F_i) + 1$$

$$b_i = ld(F_i)$$

$$\psi_i = \psi_{i-1}(\frac{b_i}{\psi_{i-1}})^{\triangle_i - 1}.$$

-Para $i = 3, \cdots, n-1$,

$$\beta_{i+1} = (-1)^{\triangle_i}(\psi_{i-1})^{\triangle_i - 1} b_{i-1}.$$

$-PRemainder(F_{n-1}, F_n) = 0$.

La secuencia de polinomios $F_1, F_2, \cdots, F_n$ se llama secuencia del resto polinomial subresultante.

En la definición vemos que hay denominadores, en los artículos de Collins y de Brown y Traub se demuestra que esas operaciones se mantienen en el anillo, es decir, que el denominador divide a los coeficientes del numerador, tanto en la expresión de ψ como de F_i.

Ahora vamos a ver la relación de la sucesión de restos polinomiales de Collins con la resultante y subresultantes.

2.5. Subresultantes

Definición 2.5.1(Subresultante): Sea K un anillo conmutativo y sean $A(x), B(x) \in K[x]$ dos polinomios con grado m, n respectivamente, ambos positivos.

$A(x) = a_0 + a_1 x + \cdots + a_m x^m$, con $deg(A(x)) = m > 0$ y

$B(x) = b_0 + b_1 x + \cdots + b_n x^n$, con $deg(B(x)) = n > 0$,

y sea $\alpha = min(m, n)$ y $\mu = max(m, n) - 1$.

Para todo $0 \leq i < \alpha$, la i^{th} subresultante de A y B se define como sigue:

1) La 0^{th} subresultante es la resultante de los polinomios A y B.

$SubRes_0(A,B) = Res(A,B) =$

$$= \begin{vmatrix} a_m & \cdots & a_1 & a_0 & & & \\ & \ddots & \ddots & \ddots & \ddots & & \\ & & a_m & \cdots & a_1 & a_0 & \\ & & & a_m & \cdots & a_1 & a_0 \\ b_n & \cdots & b_1 & b_0 & & & \\ & \ddots & \ddots & \ddots & \ddots & & \\ & & b_n & \cdots & b_1 & b_0 & \\ & & & b_n & \cdots & b_1 & b_0 \end{vmatrix}$$

$$= \begin{vmatrix} a_m & \cdots & a_1 & a_0 & & & x^{n-1}A(x) \\ & \ddots & \ddots & \ddots & \ddots & & \vdots \\ & & a_m & \cdots & a_1 & a_0 & xA(x) \\ & & & a_m & \cdots & a_1 & A(x) \\ b_n & \cdots & b_1 & b_0 & & & x^{m-1}B(x) \\ & \ddots & \ddots & \ddots & \ddots & & \vdots \\ & & b_n & \cdots & b_1 & b_0 & xB(x) \\ & & & b_n & \cdots & b_1 & b_0 \end{vmatrix}$$

que se obtiene al multiplicar la i-ésima columna por x^{m+n-i} y sumarlo a la última columna de la matriz, .

2) Para todo i, $0 < i < \alpha$, la i^{th} subresultante es:

$SubRes_i(A,B) =$

$$= \begin{vmatrix} a_m & \cdots & a_1 & a_0 & & & x^{n-i-1}A(x) \\ & \ddots & \ddots & \ddots & \ddots & & \vdots \\ & & a_m & \cdots & a_{i+1} & a_i & xA(x) \\ & & & a_m & \cdots & a_{i+1} & A(x) \\ b_n & \cdots & b_1 & b_0 & & & x^{m-i-1}B(x) \\ & \ddots & \ddots & \ddots & \ddots & & \vdots \\ & & b_n & \cdots & b_{i+1} & b_i & xB(x) \\ & & & b_n & \cdots & b_{i+1} & B(x) \end{vmatrix}$$

Ejemplos: Para $m = 5$, $n = 3$ e $i = 1$, tenemos que

$SubRes_1(A,B) =$

$$\begin{vmatrix} a_5 & a_4 & a_3 & a_2 & a_1 & xA(x) \\ 0 & a_5 & a_4 & a_3 & a_2 & A(x) \\ b_3 & b_2 & b_1 & b_0 & 0 & x^3B(x) \\ 0 & b_3 & b_2 & b_1 & b_0 & x^2B(x) \\ 0 & 0 & b_3 & b_2 & b_1 & xB(x) \\ 0 & 0 & 0 & b_3 & b_2 & B(x) \end{vmatrix}$$

Para $m = 5$, $n = 3$ e $i = 2$, tenemos que

$SubRes_2(A,B) =$

$$\begin{vmatrix} a_5 & a_4 & a_3 & A(x) \\ b_3 & b_2 & b_1 & x^2B(x) \\ 0 & b_3 & b_2 & xB(x) \\ 0 & 0 & b_3 & B(x) \end{vmatrix}$$

Utilizando transformaciones elementales sobre las columnas también obtenemos:

$SubRes_i(A,B) =$

$$= \begin{vmatrix} a_m & \cdots & a_1 & a_0 & & & 0 \\ & \ddots & \ddots & \ddots & \ddots & & \vdots \\ & & a_m & \cdots & a_{i+1} & a_i & \sum_{j=0}^{i-1} x^{j+1}a_j \\ & & & a_m & \cdots & a_{i+1} & \sum_{j=0}^{i} x^j a_j \\ b_n & \cdots & b_1 & b_0 & & & 0 \\ & \ddots & \ddots & \ddots & \ddots & & \vdots \\ & & b_n & \cdots & b_{i+1} & b_i & \sum_{j=0}^{i-1} x^{j+1}b_j \\ & & & b_n & \cdots & b_{i+1} & \sum_{j=0}^{i} x^j b_j \end{vmatrix}$$

$= DetPol(x^{n-i-1}A, \cdots, xA, A, x^{m-i-1}B, \cdots, xB, B).$

es decir, si

$$M_i = \begin{pmatrix} a_m & \cdots & a_0 & \\ \ddots & & \ddots & \\ & a_m & \cdots & a_0 \\ b_n & \cdots & b_0 & \\ \ddots & & \ddots & \\ & b_n & \cdots & b_0 \end{pmatrix}, \tag{7}$$

es la matriz obtenida de la matriz de Sylvester de A y B, quitando,

a) Las primeras i filas correspondientes a A,

b) Las primeras i filas correspondientes a B,

c) Las primeras i columnas, luego

$SubRes_i(A,B) = DetPol(M_i) = det(M_i^{m+n-2i})x^i + \cdots + det(M_i^{m+n-i})$.

Vemos que la primera igualdad se cumple con un ejemplo, supongamos que $m = 5$, $n = 3$ e $i = 1$. Tenemos que,

$$DetPol(M_1) = \begin{vmatrix} a_5 & a_4 & a_3 & a_2 & a_1 & 0 \\ 0 & a_5 & a_4 & a_3 & a_2 & a_0 \\ b_3 & b_2 & b_1 & b_0 & 0 & 0 \\ 0 & b_3 & b_2 & b_1 & b_0 & 0 \\ 0 & 0 & b_3 & b_2 & b_1 & 0 \\ 0 & 0 & 0 & b_3 & b_2 & 0 \end{vmatrix} = +\mathrm{x} \cdot \begin{vmatrix} a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \\ 0 & a_5 & a_4 & a_3 & a_2 & a_1 \\ b_3 & b_2 & b_1 & b_0 & 0 & 0 \\ 0 & b_3 & b_2 & b_1 & b_0 & 0 \\ 0 & 0 & b_3 & b_2 & b_1 & b_0 \\ 0 & 0 & 0 & b_3 & b_2 & b_1 \end{vmatrix}$$

$$= \begin{vmatrix} a_5 & a_4 & a_3 & a_2 & a_1 & xa_0 \\ 0 & a_5 & a_4 & a_3 & a_2 & xa_1 + a_0 \\ b_3 & b_2 & b_1 & b_0 & 0 & 0 \\ 0 & b_3 & b_2 & b_1 & b_0 & 0 \\ 0 & 0 & b_3 & b_2 & b_1 & xb_0 \\ 0 & 0 & 0 & b_3 & b_2 & xb_1 + b_0 \end{vmatrix}$$

que es lo mismo que $SubRes_i(A,B)$.

Por tanto tenemos que $deg(SubRes_i) \leq i$, dónde se cumple la igualdad si M_i^{m+n-2i} es no singular.

Definición 2.5.2: El coeficiente líder nominal de $SubRes_i(A,B)$, M_i^{m+n-2i}, será el i^{th} coeficiente principal de la subresultante:

$$PSC_i(A,B) = det(M_i^{m+n-2i}).$$

Sea K un anillo conmutativo y sean $A(x), B(x) \in K[x]$, dos polinomios univariantes con grados positivos m y n, respectivamente y sea:

$\lambda = min(m,n)$ y $\mu = max(m,n) - 1$.

Vamos a extender la definición de la i^{th} subresultante de A y B, para todo i, con $0 \leq i < \mu$ como sigue:

Caso 1: $(0 \leq i < \lambda)$

$SubRes_i(A,B) =$

$$= \begin{vmatrix} a_m & \cdots & a_1 & a_0 & & & x^{n-i-1}A(x) \\ & \ddots & \ddots & \ddots & \ddots & & \vdots \\ & & a_m & \cdots & a_{i+1} & a_i & xA(x) \\ & & & a_m & \cdots & a_{i+1} & A(x) \\ b_n & \cdots & b_1 & b_0 & & & x^{m-i-1}B(x) \\ & \ddots & \ddots & \ddots & \ddots & & \vdots \\ & & b_n & \cdots & b_{i+1} & b_i & xB(x) \\ & & & b_n & \cdots & b_{i+1} & B(x) \end{vmatrix}$$

Caso 2: $(i = \lambda)$

Como hemos asumido que $\lambda < \mu$, entonces $|m-n|-1 > 0$, por lo que tenemos, o bien $m > n+1$, o bien $n > m+1$, con $\lambda = n$, o $\mu = m$, respectivamente. Los dos casos son simétricos. Supongamos que $\lambda = n$, luego

$SubRes_\lambda(A,B) =$

$$= det \begin{vmatrix} b_n & \cdots & & & x^{m-n-1}B(x) \\ & \ddots & & & \vdots \\ & & b_n & \cdots & xB(x) \\ & & & & B(x) \end{vmatrix}$$

Por tanto, $SubRes_\lambda(A,B) = b_n^{m-n-1} \cdot B(x) = ld(B)^{m-n-1} \cdot B$.

En el caso de que $\lambda = m$, tenemos $SubRes_\lambda(A,B) = a_m^{n-m-1} \cdot A(x) = ld(A)^{n-m-1} \cdot A$.

Podemos escribir entonces, $SubRes_\lambda(A,B) = ld(C)^{|m-n|-1} \cdot C$, donde

$$C = \begin{cases} A, & ,if deg(B) > deg(A)+1 \\ B, & ,if deg(A) > deg(B)+1 \end{cases}$$

Caso 2: $(\lambda < i < \mu)$

$$SubRes(A,B) = 0.$$

Definición 2.5.3: Una subresultante S_i se dice que es ***defectuosa de grado*** r si:

$$r = Deg(S_i) < i.$$

en otro caso, se dice que S_i es regular.

Vamos a ver una definición alternativa de las subresultantes.

Proposición 2.5.4: Sea K un anillo conmutativo y sean $A(x), B(x) \in K[x]$ dos polinomios con grados positivos m y n respectivamente y además,

$$\lambda = min(m, n) \text{ y } \mu = max(m, n) - 1.$$

La i^{th} subresultante de A y B, para todo i con $(0 \leq i < \mu)$, está dado por:

Caso 1: $(0 \leq i < \lambda)$

$SubRes_i(A, B) = DetPol(x^{n-i-1}A, \cdots, xA, A, x^{m-i-1}B, \cdots, xB, B).$

Caso 2: $(i = \lambda)$

$SubRes_\lambda(A, B) =$

$$= \begin{cases} DetPol(x^{n-m-1}A, \cdots, xA, A) = ld(A)^{n-m-1} \cdot A, & , if deg(B) > deg(A) + 1 \\ DetPol(x^{m-n-1}B, \cdots, xB, B) = ld(B)^{m-n-1} \cdot B, & , if deg(A) > deg(B) + 1 \end{cases}$$

Caso 3: $(\lambda < i < \mu)$

$$SubRes_i(A, B) = 0.$$

Lema 2.5.5: Sea K un anillo conmutativo y $A(x), B(x) \in K[x]$ polinomios con grados positivos m y n respectivamente en el anillo K, y sean

$$\lambda = min(m, n) \text{ y } \mu = max(m, n) - 1.$$

Luego para todo i, con $(0 \leq i < \mu)$,

$$SubRes_i(A, B) = (-1)^{(m-i)(n-i)} SubRes_i(B, A).$$

Demostración:

1. Para todo $0 \leq i < \lambda$,

$$Matrix(x^{m-i-1}B, \cdots, xB, B, x^{n-i-1}A, \cdots, xA, A)$$

la podemos obtener de la matriz

$$Matrix(x^{n-i-1}A, \cdots, xA, A, x^{m-i-1}B, \cdots, xB, B),$$

utilizando $(m - i)(n - i)$ trasposiciones de filas.

2. Para $i = \lambda$ luego $(m - n)(n - n) = 0$ y

$$SubRes_i(A, B) = SubRes_i(B, A).$$

3. Para finalizar, para todo $\lambda < i < \mu$,

$$SubRes_i(A, B) = SubRes_i(B, A) = 0.$$

□

Lema 2.5.6: Sea K un anillo conmutativo, y sea $A(x), B(x) \in K[x]$ polinomios con grados positivos m y n respectivamente con coeficientes en el anillo K, $m \geq n > 0$. Luego,

$$\lambda = min(m, n) \text{ y } \delta = m - n + 1.$$

Luego,

$$\begin{aligned} b_n^\delta SubRes_{\lambda-1}(A, B) &= b_n^\delta DetPol\left(A, x^{m-n}B, x^{m-n-1}B, \ldots, xB, B\right) \\ &= (-1)^{m-n+1} DetPol\left(x^{m-n}B, \ldots, xB, B, b_n^\delta A\right) \\ &= (-1)^\delta b_n^\delta PRemainder(A, B). \end{aligned}$$

Especificamente, si K es un dominio íntegro, tendríamos que,

$$PRemainder(A, B) = (-1)^{m-n+1} SubRes_{n-1}(A, B).$$

Demostración:

$$\begin{aligned} b_n^\delta SubRes_{\lambda-1}(A, B) &= b_n^\delta DetPol\left(A, x^{m-n}B, x^{m-n-1}B, \ldots, xB, B\right) \\ &\quad \text{(Definición } SubRes\text{)} \\ &= (-1)^{m-n+1} DetPol\left(x^{m-n}B, \ldots, xB, B, b_n^\delta A\right) \\ &\quad \text{(Propiedades + Transposiciones)} \\ &= (-1)^\delta b_n^\delta PRemainder(A, B). \\ &\quad \text{(Teorema 2.3.4). } \square \end{aligned}$$

2.6. Subresultantes y grado del máximo común divisor

Lema 2.6.1: Sea K un anillo conmutativo, y sean $A(x), B(x) \in K[x]$ polinomios, con grados positivos m y n respectivamente con coeficientes en el anillo K. Luego existen polinomios $T_i(x)$ y $U_i(x) \in K[x]$ tal que para todo $0 \leq i < max(m, n) - 1$,

$$A(x) \cdot T_i(x) + B(x) \cdot U_i(x) = SubRes_i(A, B),$$

donde,

$deg(T_i) < deg(B) - i = n - i$ y $deg(U_i) < deg(A) - i = m - i$.

Demostración: El lema es trivial en el caso para todo $min(m,n) \leq i < max(m,n) - 1$. Por tanto estudiaremos el caso: $0 \leq i < \lambda = min(m,n)$.

Extendamos la matriz i^{th} Sylvester con la última columna y obtenemos:

$$P_i = \begin{vmatrix} a_m & \cdots & a_1 & a_0 & & & x^{n-i-1}A(x) \\ & \ddots & \ddots & \ddots & \ddots & & \vdots \\ & & a_m & \cdots & a_{i+1} & a_i & xA(x) \\ & & & a_m & \cdots & a_{i+1} & A(x) \\ b_n & \cdots & b_1 & b_0 & & & x^{m-i-1}B(x) \\ & \ddots & \ddots & \ddots & \ddots & & \vdots \\ & & b_n & \cdots & b_{i+1} & b_i & xB(x) \\ & & & b_n & \cdots & b_{i+1} & B(x) \end{vmatrix}$$

Luego, (Desarrollando por la última columna)

$SubRes_i(A,B)$

$$\begin{aligned} &= x^{n-i-1}A(x) \cdot P_{1,m+n-2i} + \cdots + A(x) \cdot P_{n-i,m+n-2i} \\ &\quad + x^{m-i-1}B(x) \cdot P_{n-i+1,m+n-2i} + \cdots + B(x) \cdot P_{m+n-2i,m+n-2i} \end{aligned}$$

(Tomando factor común)

$$\begin{aligned} &= A(x)(P_{1,m+n-2i}x^{n-i-1} + \cdots + P_{n-i,m+n-2i}) \\ &\quad + B(x)(P_{n-i+1,m+n-2i}x^{m-i-1} + \cdots + P_{m+n-2i,m+n-2i}) \\ &= A(x) \cdot T_i(x) + B(x) \cdot U_i(x). \end{aligned}$$

Los coeficientes de $T_i(x)$ y $U_i(x)$ son cofactores de la última columna de P_i y por tanto elementos de K:

$deg(T_i) < deg(B) - i = n - i$ y $deg(U_i) < deg(A) - i = m - i$. □

Lema 2.6.2: Sean $A(x)$ y $B(x)$ polinomios con grados positivos m y n respectivamente, en un dominio íntegro K. Asumimos que existen polinomios $T_i(x)$ y $U_i(x)$, no ambos cero en K, tal que para todo $0 \leq i < max(m,n) - 1$,

$$A(x) \cdot T_i(x) + B(x) \cdot U_i(x) = 0,$$

con,

$$deg(T_i) < n - i \text{ y } deg(U_i) < m - i.$$

Luego, $SubRes_i(A, B) = 0$.

Lema 2.6.3: Sean $A(x)$ y $B(x)$ polinomios con grados positivos m y n respectivamente, en un dominio íntegro K. Luego, para todo $0 \leq i < max(m, n) - 1$, el i^{th} coeficiente principal nominal subresultante de A y B desaparece, es decir,

$$PSC_i(A, B) = 0$$

si y sólo si existen polinomios $T_i(x)$,$U_i(x)$ y $C_i(x)$ no todos cero en K tal que,

$$A(x) \cdot T_i(x) + B(x) \cdot U_i(x) = C_i(x),$$

donde

$$deg(T_i) < n - i,\ deg(U_i) < m - i \text{ y } deg(C_i) < i.$$

Lema 2.6.4: Sea K un dominio de factorización única, y $A(x), B(x)$ polinomios con grados positivos m y n respectivamente, con coeficientes en K. Luego, para todo $0 \leq i < min(m, n)$:

$$A(x) \cdot T_i(x) + B(x) \cdot U_i(x) = 0,$$

donde, $deg(T_i) < n - i$ y $deg(U_i) < m - i$, si y sólo si $A(x)$ y $B(x)$ tienen un divisor común de grado $> i$.

Demostración: Sea $D(x)$ el común divisor de $A(x)$ y $B(x)$ de grado más alto entre todos ellos. Por tanto se tiene: $A(x) = U'(x)D(x)$ y $B(x) = T'(x)D(x)$,

donde hemos asumido que $U'(x), T'(x)$ no tienen un común divisor no constante. Es claro que, $deg(U') = m - deg(D)$ y $deg(T') = n - deg(D)$.

$\Longleftarrow$) Supongamos que $deg(D) > i$, luego escojamos $T_i = T'$ y $U_i = -U'$. Por tanto,

$$A(x) \cdot T'(x) - B(x) \cdot U'(x) = 0,$$

donde, $deg(T') < n - i$ y $deg(-U') < m - i$.

$\Longrightarrow$) Como,

$$A(x) \cdot T_i(x) + B(x) \cdot U_i(x) = 0,$$

también tenemos,

$U'(x) \cdot T_i(x) + T'(x) \cdot U_i(x) = 0$ o $U'(x) \cdot T_i(x) = -T'(x) \cdot U_i(x)$.

Ahora, como $U'(x)$ y $T'(x)$ no tienen un común divisor no constante, cada divisor de $U'(x)$ tiene que estar asociado con un divisor de $U_i(x)$, es decir,

$$deg(U') \leq deg(U_i) < m - i.$$

Es decir,

$$deg(U') = m - deg(D) < m - i \rightarrow deg(D) > i. \square$$

Lema 2.6.5: Sea K un dominio de factorización única, y sean $A(x)$ y $B(x)$ los polinomios con grados positivos m y n, respectivamente con coeficientes en K. Luego para todo $0 \leq i < min(m, n)$, las siguientes afirmaciones son equivalentes:

1. $A(x)$ y $B(x)$ tienen un divisor común de grado $> i$;

2. $(\forall j \leq i)$ $[SubRes_j(A, B) = 0]$;

3. $(\forall j \leq i)$ $[PSC_j(A, B) = 0]$.

Demostración:

1) $\Longrightarrow$ 2) Como A y B tienen un divisor común de grado $> i$, entonces para todo $j \leq i$,

$$A(x) \cdot T_j(x) + B(x) \cdot U_j(x) = 0,$$

donde $deg(T_j) < n-j$ y $deg(U_j) < m-i$, lo que implica que, $(\forall j \leq i)$ $[SubRes_j(A, B) = 0]$. Por el Lema 2.5.5.

2) $\Longrightarrow$ 3) Trivial.

3) $\Longrightarrow$ 1) Se realiza por inducción para todo $j \leq i$.

Para $j = 0$, es claro que $PSC_0(A, B) = 0$ implica que $Resultant(A, B) = 0$, y que A y B tienen un divisor común de grado > 0.

Suponemos que es cierto para $j - 1$ y lo probamos para j.

$PSC_j(A, B) = 0$ y A y B tiene un divisor común de grado $> j - 1$, esto implica que

$$(\exists C_j(x), deg(C_j) < j)[A(x) \cdot T_j(x) + B(x) \cdot U_j(x) = C_j(x)]$$

con, $deg(T_j) < n - j$, $deg(U_j) < m - j$.

Pero como A y B son divisibles por un polinomio de grado $\geq j$, esto implica que $C_j(x) = 0$.

Por tanto, $A(x) \cdot T_j(x) + B(x) \cdot U_j(x) = 0$, con $deg(T_j) < n - j$, $deg(U_j) < m - j$.

Luego implica que A y B tienen un divisor común de grado $> j$. $\square$

Colorario 2.6.6: Sea K un dominio de factorización única, y sean $A(x)$y $B(x)$ dos polinomios con grados positivos m y n respectivamente, con coeficientes en K. Luego para todo $0 < i \leq min(m, n)$, las siguientes afirmaciones son equivalentes:

1. $A(x)$ y $B(x)$ tienen un divisor común de grado $= i$;

2. $(\forall j < i)$ $[SubRes_j(A, B) = 0] \wedge SubRes_i(A, B) \neq 0$;

3. $(\forall j < i)\ [PSC_j(A,B) = 0] \wedge PSC_i(A,B) \neq 0$.

Corolario 2.6.7: Sea K un anillo conmutativo, y sean $A(x)$ y $B(x)$ dos polinomios con grados positivos m y n, respectivamente, con coeficientes en el anillo K.

$$A(x) = a_m x^m + a_{m-1}x^{m-1} \cdots + a_0, \text{ y}$$
$$B(x) = b_n x^n + b_{n-1}x^{n-1} \cdots + b_0,$$

donde $deg(A) = m > 0$ y $deg(B) = n > 0$.

Si $deg(\phi(A)) = m$ y $deg(\phi(B)) = k$, $(0 \leq k \leq n)$,

luego para todo $0 \leq i < max(m,k) - 1$,

$$\phi(SubRes_i(A,B)) = \phi(a_m)^{n-k} SubRes_i(\phi(A), \phi(B).$$

2.7. Cadenas de subresultantes

Definición 2.7.1: Sean S un anillo conmutativo y sean $A(x)$ y $B(x) \in S[x]$ dos polinomios univariantes con grados n_1 y n_2 respectivamente, con $n_1 \geq n_2$:

$A(x) = a_{n_1}x^{n_1} + a_{n_1-1}x^{n_1-1} + \cdots + a_0$, con $deg(A) = n_1 > 0$, y

$B(x) = b_{n_2}x^{n_2} + b_{n_2-1}x^{n_2-1} + \cdots + b_0$, con $deg(B) = n_2 > 0$.

Sea,

$$n = \begin{cases} n_1 - 1, & si \quad n_1 > n_2, \\ n_2, & si \quad n_1 = n_2 \end{cases}$$

La cadena subresultante de A y B viene dada por:

$\langle S_{n+1} = A, S_n = B, S_{n-1} = SubRes_{n-1}(A,B), \cdots, S_1 = SubRes_1(A,B), S_0 = SubRes_0(A,B)\rangle$.

La cadena de los coeficientes principales de A y B viene dada por:

$\langle PSC_{n+1} = 1, PSC_n = CoefLid(S_n), PSC_{n-1} = CoefLid(S_{n-1}), \cdots, PSC_1 = CoefLid(S_1)$
$, PSC_0 = CoefLid(S_0)\rangle$.

Con $CoefLid$ denotamos al coeficiente asociado al término de mayor grado.

Definición 2.7.2: Una cadena subresultante se dice que es defectuosa si alguno de sus elementos es defectuoso, es decir, si para algún $(0 \leq i \leq \mu)$,$(\mu = max(n_1, n_2) - 1)$

$$r = deg(S_i) < i;$$

en caso contrario diremos que es regular.

Nuestro objetivo principal es obtener una relación entre las cadenas subresultantes y la secuencia del resto polinomial subresultante. Esta relación se describe con precisión más tarde en el teorema de la cadena subresultante. No proporcionaremos las demostraciones y nos limitaremos a describir dicha relación. La referencia en todo lo que sigue en el libro Mishra que está en la bibliografía.

Lema 2.7.3: Se demuestra que una cadena de subresultantes puede ser dividida en bloques de subresultantes.

$$\langle S_i, S_{i-1}, \cdots, S_{j+1}, S_j \rangle, \qquad n+1 \geq i \geq j \geq 0,$$

con las siguientes propiedades:

1. O bien es el bloque formado por ceros, es decir, Si $j = 0$ y $S_i = S_{i-1} = \cdots = S_{j+1} = S_j = 0$, esto es llamado el bloque cero. Además se demuestra que $S_{i-1} \neq 0$.

2. O bien $S_i \neq 0$, $S_j \neq 0$, $S_i \backsim S_j$ y $S_{i-1} = \cdots = S_{j+1} = 0$. Esto es llamado los bloques que no son cero. Es este caso se demuestra que S_j es siempre regular y si $i > j$, S_i es defectuosa.

Escribireos los bloques que no son cero como sigue:

$$\langle \bar{S}_0, \bar{S}_1, \cdots, \bar{S}_i \rangle.$$

El primer polinomio subresultante en el i^{th} bloque que no es cero se llama elemento superior, y la última subresultante se llama elemento inferior. Llamaremos al elemento superior $\bar{S}_i^{top}$ y al elemento inferior $\bar{S}_i^{inf}$. El PSC, $\bar{R}_i^{inf}$, se define similar.

Ejemplo 1:

Vamos a ver un ejemplo de una configuración válida.

S_i	S_{10}	S_9	S_8	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0
deg	10	7	-∞	7	6	3	-∞	3	0	0	0

Supongamos que tengamos $A(x)$, $B(x) \in S[x]$ dos polinomios con $deg(A) = 10$ y con $deg(B) = 7$. Por lo tanto tendríamos que S_{10} tiene $deg(S_{10}) = 10$ y que S_9 tiene $deg(S_9) = 7$, ya que por la definición 2.7.1 tendríamos que $n = deg(A) - 1 = 10 - 1 = 9$ y la cadena vendría dada por:

$\langle S_{10} = A, S_9 = B, S_{9-1} = SubRes_{9-1}(A, B), \cdots, S_1 = SubRes_1(A, B), S_0 = SubRes_0(A, B) \rangle$.

Por la propiedad (2) vista anteriormente como $S_{10} \neq 0$ y $S_9 \neq 0$, entonces $S_8 = 0$ y por tanto $deg(S_8) = -\infty$, por convenio.

S_7 podría tener $deg(S_7) = 7$ y por tantp S_7 sería regular y S_9, S_8 y S_7 cumplirían las propiedades.

Supongamos que para los demás $deg(S_6) = 6$, $deg(S_5) = 3$, $deg(S_4) = -\infty$, $deg(S_3) = 3$, y los demás tienen grado 0, todos ellos cumplen las propiedades anteriores.

Ahora vamos a ver los bloques que no son cero de este ejemplo.

Serían $\langle \bar{S}_0 = S_{10}, \bar{S}_1 = (S_9, S_8, S_7), \bar{S}_2 = S_6, \bar{S}_3 = (S_5, S_4, S_3)\rangle$, donde S_2, S_1 y S_0 no aparecen al ser los bloques cero.

Ahora vamos a ver los elementos superior e inferior de los bloques que no son cero:

$\bar{S}_0^{top} = \bar{S}_0^{inf} = S_{10}$, $\bar{S}_1^{top} = S_9$, $\bar{S}_1^{inf} = S_7$, $\bar{S}_2^{top} = \bar{S}_2^{inf} = S_6$ y $\bar{S}_3^{top} = S_5$, $\bar{S}_3^{inf} = S_3$.

Ejemplo 2:

Ahora vamos a ver un ejemplo de configuración imposible:

S_i	S_4	S_3	S_2	S_1	S_0
deg	4	2	2	0	0

Supongamos que tengamos $A(x)$, $B(x) \in S[x]$ dos polinomios con $deg(A) = 4$ y con $deg(B) = 2$. Por lo tanto tendríamos que S_4 tiene $deg(S_4) = 4$ y que S_3 tiene $deg(S_3) = 2$, ya que por la definición 2.7.1 tendríamos que $n = deg(A) - 1 = 4 - 1 = 3$ y la cadena vendría dada por:

Si tuviéramos que $deg(S_2) = 2$ y que $deg(S_1) = -\infty$ y $deg(S_0) =$, entonces sería imposible ya que por la propiedad 2, al tener los mismos grados S_2 y S_3 tendríamos que tener que $deg(S_4) = -\infty$.

Ejemplo 3:

En este tercer ejemplo realizado con la ayuda del programa Maple.

S_i	S_{13}	S_{12}	S_{11}	S_{10}	S_9	S_8	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0
deg	13	11	11	7	-∞	-∞	7	6	5	4	3	-∞	-∞	-∞

He calculado los S_i para poder ver un ejemplo real y poder mostrarlo. El ejemplo sería el siguiente: Suponiendo que variable líder es la y. Los dos polinomios son $A(x, y) = ((y^2 + 6)(x-1)-y(x^2+1)+y^{10})(-y^3+x^2+xy)$ y $B(x) = ((x^2+6)(y-1)-x(y^2+1)+xy^8)(-y^3+x^2+xy)$, con $deg(A) = 13$ y $deg(B) = 11$. Los resultados al calcular el subresultante de cada uno da los siguientes resultados sobre los grados: $[-\infty, -\infty, -\infty, 3, 4, 5, 6, 7, -\infty, -\infty, 7, 11, 13]$. Dónde empieza por S_0 y va subiendo hasta llegar a S_{12}, observamos que hay un problema, de la forma que lo hemos definido al calcular S de mayor tamaño sería 13 y sólo nos sale

13, pero eso es debido a que Maple no calcula la subresultante 11 ya que no la tiene definida, pero nosotros si la hemos definido en la proposición 2.5.4. Por tanto ya tendríamos calculado un ejemplo real.

Teorema 2.7.4:(Teorema de la cadena subresultante) Sea S un dominio integral y sea

$$\langle S_{n+1}, S_n, \cdots, S_0 \rangle$$

una cadena subresultante de S_{n+1} y S_n en $S[x]$ $(deg(S_{n+1}) \geq deg(S_n))$.

1. Para $j = 1, \cdots, n$, si S_{j+1} y S_j son ambos regulares entonces,

$$(-R_{j+1})^2 S_{j-1} = PRemainder(S_{j+1}, S_j).$$

2. Para $j = 1, \cdots, n$, si S_{j+1} es regular y S_j es defectuosa de grado r $(r < j)$, entonces

$$S_{j-1} = S_{j-2} = \cdots = S_{r+1} = 0,$$
$$(R_{j+1})^{j-r} S_r = CoefLid(S_j)^{j-r} S_j, \quad r \geq 0,$$
$$(-R_{j+1})^{j-r+2} S_{r-1} = PRemainder(S_{j+1}, S_j), \quad r \geq 1.$$

Corolario 2.7.5: Sea S un dominio integral y sea

$$\langle \bar{S}_0, \bar{S}_1, \cdots, \bar{S}_i \rangle$$

una secuencia de bloques que no son ceros de una cadena subresultante de S_{n+1} y S_n en $S[x]$ $(deg(S_{n+1})) \geq deg(S_n)$. Por tanto,

$$(\bar{R}^{inf}_{(i-1)})^{\delta_{i+1}-2} \bar{S}^{inf}_i = CoefLid(\bar{S}^{top}_i)^{\delta_{i+1}-2} \bar{S}^{top}_i,$$
$$(-\bar{R}^{inf}_{(i-1)})^{\delta_{i+1}} \bar{S}^{top}_{(i+1)} = PRemainder(\bar{S}^{inf}_{(i-1)}, \bar{S}^{top}_i).$$

Nota 2.7.6: Podemos ver que,

$$\bar{S}^{inf}_i \backsim \bar{S}^{top}_i \quad \text{y}$$
$$\bar{S}^{top}_{(i+1)} \backsim PRemainder(\bar{S}^{inf}_{(i-1)}, \bar{S}^{inf}_i)$$
$$\Longrightarrow$$
$$\bar{S}^{inf}_{(i+1)} \backsim PRemainder(\bar{S}^{inf}_{(i-1)}, \bar{S}^{inf}_i).$$

Corolario 2.7.7: Sea S un dominio integral, y sea $F_1(x)$ y $F_2(x) \in S[x]$ $(deg(F_1) \geq deg(F_2))$. Ahora consideramos la secuencia de restos de polinomios: $F_1, F_2, \cdots, F_k$ y su cadena subresultante, con la siguiente secuencia de bloques que no son ceros:

$$\langle \bar{S}_0, \bar{S}_1, \cdots, \bar{S}_l \rangle.$$

Luego los elementos de la secuencia de restos de polinomios es similar a las subresultantes regulares, en su respectivo orden, es decir,

1. $k = l + 1$.

2. $\bar{S}_i^{inf} \backsim F_{i+1}$.

Teorema 2.7.8: Sea S un dominio de factorización única,y sea $F_1(x), F_2(x) \in S[x]$ con $deg(F_1) \geq deg(F_2)$.

Ahora consideramos la secuencia del resto polinomial subresultante: $F_1, F_2, \cdots, F_k$, y su cadena subresultante, con la siguiente secuencia de bloques que no son ceros:

$$\langle \bar{S}_0, \bar{S}_1, \cdots, \bar{S}_l \rangle.$$

Por tanto

1. $F_{i+1} = \bar{S}_i^{top}$, $i = 0, \cdots, k-1$.

2. $\psi_{i+1} = \bar{R}_i^{inf}$, $i = 0, \cdots, k-1$.

Observación: Si nos fijamos en el anterior teorema tenemos que,

$F_1 = \bar{S}_0^{top}$.

$F_2 = \bar{S}_1^{top} \sim \bar{S}_1^{inf}$ es regular, por tanto si suponemos que $\bar{S}_1^{inf} = S_{j_1}$, entonces $deg(F_2) = j_1$.

$F_3 = \bar{S}_2^{top} \sim \bar{S}_2^{inf}$ es regular, por tanto si suponemos que $\bar{S}_2^{inf} = S_{j_2}$, entonces $deg(F_3) = j_2$.

Siguiendo este proceso llegamos a que $deg(F_k) = j_k$ que es el $deg(mcd(A, B))$.

Teorema 2.7.9: Sea,

$$\langle S_{n+1}, S_n, \cdots, S_0 \rangle$$

una cadena subresultante de S_{n+1} y S_n en $S[x]$. 1. Para $j = 1, \cdots, n$, si S_{j+1} y S_j son regulares, entonces

$$(-R_{j+1})^2 \cdot S_{j-1} = PRemainder(S_{j+1}, S_j).$$

2. Para $j = 1, \cdots, n$, si S_{j+1} es regular y S_j es defectuosa de grado r, $r < j$ luego,

$$S_{j-1} = S_{j-2} = \cdots = S_{r+1} = 0,$$
$$(R_{j+1})^{j-r} S_r = CoefLid(S_j)^{j-r} S_j, r \geq 0,$$
$$(-R_{j+1})^{j-r+2} S_{r-1} = PRemainder(S_{j+1}, S_j), r \geq 1.$$

Capitulo 3

3. Descomposición algebraica de Thomas

3.1. Preliminares

En este trabajo veremos la introducción de los conceptos de sistemas simples y la descomposición de Thomas para sistemas algebraicos. Al final veremos una construcción para la descomposición de Thomas.

Ejemplo 3.1.1:

Consideramos la ecuación,

$$p = x^3 + (3y+1)x^2 + (3y^2+2y)x + y^3 = 0$$

Una descomposición de Thomas viene dada por:

$$S_1 = x^3 + (3y+1)x^2 + (3y^2+2y)x + y^3 = 0, 27y3 - 4y = 0$$

y

$$S_2 = 6x^2 + (-27y2 + 12y + 6)x - 3y2 + 2y = 0, 27y3 - 4y = 0$$

3.2. Definiciones y conceptos previos

Sea K un cuerpo computable de característica 0 (Es decir, no existe n tal que $1_R + ..nsumandos + .. + 1_R = 0$) y $R := K[x_1, x_2, \ldots, x_n]$ el anillo de polinomios en n variables. Si todas las operaciones en K son calculables, llamamos a K cuerpo computable.

Definición 3.2.1: Un orden total $<$ en $\{1, x_1, x_2, \ldots, x_n\}$ con $1 < x_i$, $\forall i$ se llama ***ranking***.

Observación 3.2.2: Asumimos que $i < j$ implica que $x_i < x_j$.

Definición 3.2.3: Una variable x se llama variable *líder* de $p \in R$ si x es la variable mayor tal que p es un polinomio no constante en esa variable. En este caso escribimos $\boldsymbol{ld}(p) = x$.

Observación 3.2.4: Si $p \in K$, se dice que $ld(p) = 1$, es decir, p es una constante.

Definición 3.2.5: El grado de p en $ld(p)$ se llama ***grado principal*** de p y el coeficiente de $ld(p)$ se llama ***coeficiente líder*** de p. Grado principal se denota por ***mdep***(p) y el coeficiente líder se denota por ***init***(p).

Para $a \in \overline{K}^n$, donde $\overline{K}$ denota la clausura algebraica de K , definimos el siguiente homomorfismo:

$$\phi_a : K[x_1, \ldots, x_n] \to \overline{K},$$

$$x_i \mapsto a_i.$$

Para $a \in \overline{K}^i$, $k-1 \leq i \leq n$, definimos:

$$\phi_{<x_k,a} : K\left[x_1, \ldots, x_n\right] \to \overline{K}\left[x_k, \ldots, x_n\right] : \begin{cases} x_j \mapsto a_j, & j < k, \\ x_j \mapsto x_j, & \textit{en otro caso.} \end{cases}$$

Dado un polinomio $p \in R$, los símbolos $p_=$ y $p_{\neq}$ denotan la ecuación $p = 0$ y la inecuación $p \neq 0$, respectivamente. Utilizaremos de aquí en adelante $ld(p_=) := ld(p)$ y $ld(p_{\neq}) := ld(p)$.

Un conjunto finito de ecuaciones e inecuaciones se llama ***sistema algebraico*** sobre R. Una solución de $p_=$ o $p_{\neq}$ es una n-upla $\mathbf{a} \in \overline{K}^n$ con $\phi_a(p) = 0$ o $\phi_a(p) \neq 0$, respectivamente.

Decimos que $\mathbf{a} \in \overline{K}^n$ es solución de un sistema S si es solución de cada elemento de S.

El conjunto de todas las soluciones de S se denota por ***Sol***(S).

El conjunto de todas las ecuaciones se denotan por $S^= = \{p_=/p_= \in S\}$ y $S^{\neq} = \{p_{\neq}/p_{\neq} \in S\}$.

Definamos $S_x := \{p \in S/ld(p) = x\}$. La situación en la que sea claro que $|S_x|$=1, podemos escribir S_x para denotar al único elemento de S_x.

El subconjunto $S_{<x} := \{p \in S/ld(p) < x\}$ es un sistema sobre $K[y|y < x]$.

Ejemplos:

(1) Calcular el grado principal, el coeficiente líder y la variable líder de p. Siendo $p = 2x_1(x_2 + x_4) - x_5^2(x_3 + 1)$.

La variable líder es x_5, es decir, $ld(p) = x_5$. El grado principal de p es $mdeg(p) = 2$. El coeficiente líder de p es $init(p) = -x_3 - 1$.

(2) Calcular el grado principal, el coeficiente líder y la variable líder de S, con $x < y < z$. Siendo $p = z^2(xy + x^2 + y + 2) - y^2(xy - x^4 + 2) + z(xy + x + y + 20)$.

La variable líder es z, es decir, $ld(p) = z$. El grado principal de p es $mdeg(p) = 2$. El coeficiente líder de p es $init(p) = (xy + x^2 + y + 2)$.

3.3. Sistemas simples

Definición 3.3.1 (Sistemas simples): Sea S un sistema.

1. S es **triangular** si $|S_{x_i}| \leq 1$, $\forall 1 \leq i \leq n$ y $S \cap \{C_=, C_{\neq} | C \in K\} = \emptyset$.

2. Diremos que S **cumple la propiedad de los coefcientes líderes** si $\forall i$, $1 \leq i \leq n$, si $p \in S_{x_i}$ y $a \in Sol(S_{<x_i})$ entonces $\phi_a(init(p)) \neq 0$.

3. Diremos que S es **libre de cuadrados** si se cumple: $\forall i$, $1 \leq i \leq n$, $\forall a \in Sol(S_{<x_i})$ y $p \in S_{x_i}$ entonces $\phi_{<x_i,a}(p) \in \overline{K}[x_i]$ es libre de cuadrados.

4. Diremos que S es **simple** si es *triangular*, cumple *cumple la propiedad de los coeficientes líderes* y es *libre de cuadrados*.

Ejemplos:

(1) Veamos un ejemplo de un sistema S tal que no es triangular:

x_1 x_2 x_3 x_4

*_____*_____*_____*

$(3x_1)_=$ $(x_1 + 3)_=$

No es triangular ya que $|S_{x_1}| \geq 2$, por lo que incumple que $|S_{x_i}| \leq 1$.

(2) Veamos un ejemplo de un sistema S tal que no es libre de cuadrados:

x_1 x_2 x_3 x_4

*_____*_____*_____*

$(4)_{\neq}$ $(x_4^2 + 4x_4 + 4)_=$

No es libre de cuadrados ya que $(x_4^2 + 4x_4 + 4)_=$ tiene una raíz doble.

Lema 3.3.2: Cada sistema simple tiene al menos una solución.

Demostración: En particular, si $b \in Sol(S_{<x})$ y S_x es no vacío, entonces $\phi_{<x,b}(S_x)$ es un polinomio en una variable con exactamente $mdeg(S_x)$ diferentes raíces.

En el caso en el que S_x sea una ecuación, cada solución $b \in Sol(S_{<x})$ se extiende a una solución $(b, a) \in Sol(S_{\leq x})$ con $mdeg(S_x)$ posibles opciones de $a \in \overline{k}$.

De lo contrario, todas menos un número finito $a \in \overline{K}$ generan una solución $(b, a) \in Sol(S_{\leq x})$, porque la inecuación S_x excluye $mdeg(S_x)$ diferente a, y $S_x = \emptyset$ no impone restricciones sobre a. □

Por otra parte, si $(a_1, \ldots, a_n) \in Sol(S)$ donde S es un sistema sobre $K[x_1, \ldots, x_n]$ con $x_1, \ldots, x_n$, luego $(a_1, \ldots, a_i) \in Sol(S_{\leq x_i})$.

Se transforma un sistema en un conjunto de sistemas simples haciendo una partición del conjunto de las soluciones. Esto también se llama descomposición en sistemas simples.

Definición 3.3.3: Una familia $(S_i)_{i=1}^m$ se llama ***descomposición*** de S si $Sol(S) = \cup_{i=1}^m Sol(S_i)$. Se llama ***disjunta*** si $Sol(S_i) \cap Sol(S_i) = \emptyset$, $\forall i \neq j$. Una descomposición disjunta de un sistema en sistemas simples se llama ***descomposición de Thomas***.

Ejemplo 3.3.4: Descomposición de Thomas de $\{(p := ax^2 + bx + c)_=\} \subseteq Q[a, b, c, x]$ con $a < b < c < x$.

<u>1^a *Paso*</u>: Garantizamos que init(p) de p no es 0. Ahora insertamos $(init(p))_{\neq} = (a)_{\neq}$ en el sistema.

Luego tenemos una partición del sistema en $\{p_=, a_{\neq}\} \cup \{p_=, a_=\}$.

Ya que restringimos el conjunto de soluciones del sistema, consideramos el sistema $\{p_=, (a)_=\}$ que es equivalente a $\{(bx + c)_=, (a)_=\}$.

Volvemos a partir el sistema $\{(bx + c)_=, a_=\}$ en $\{(bx + c)_=, b_{\neq}, a_=\} \cup \{p_=, b_=, a_=\}$.

Similarmente, agregamos $(b)_{\neq}$ para asegurar $init(bx + c) \neq 0$ y obtener el caso especial simple $\{(c)_=, (b)_=, (a)_=\}$.

Tenemos tres sistemas, $\{(ax^2+bx+c)_=, (a)_{\neq}\}$, $\{(bx+c)_{\neq}, (b)_{\not=}, (a)_=\}$ y $\{(c)_=, (b)_=, (a)_=\}$, dónde se ve que el segundo y tercero son simples:

Observamos que el primer sistema no es simple ya que puede tener alguna raíz doble, ya que si por ejemplo tomamos $a = 1$, $b = 0$, $c = 0$, la solución $\in Sol(S^1_{<x})$.

Primer sistema:

x c b a

*_____*_____*_____*

$(ax^2 + bx + c)_=$ $(a)_{\neq}$

Segundo sistema:

x c b a

*_____*_____*_____*

$(bx + c)_{\neq}$ $(b)_{\neq}$ $(a)_=$

Tercer sistema:

x c b a

*_____*_____*_____*

$(c)_=$ $(b)_=$ $(a)_=$

<u>2^a *Paso*</u>: Tenemos que asegurarnos que p es libre de cuadrados por la inserción de $(4ac-b^2)_{\neq}$ en el primer sistema. Tenemos que considerar el sistema $\{(p)_=, (4ac-b^2)_=, (a)_{\neq}\}$. Como p es cuadrático en el sistema, lo podemos reemplazar por $2ax+b$. Ahora todos los sistemas son simples y hemos obtenido *la descomposión de Thomas*.

Del sistema $\{p_=, a_{\neq}\}$ obtenemos una partición $\{(p)_=, (4ac-b^2)_=, a_{\neq}\} \cup \{p_=, (4ac-b^2)_{\neq}, a_{\neq}\}$. De $(4ac-b^2)=0$ obtenemos la solución $(2ax+b)_=$ ya que $x = \frac{-b \pm \sqrt{0}}{2a} = \frac{-b}{2a}$.

x c b a

*_____*_____*_____*

$(ax^2+bx+c)_=$ $(4ac-b^2)_{\neq}$ $(a)_{\neq}$

x c b a

*_____*_____*_____*

$(2ax+b)_=$ $(4ac-b^2)_=$ $(a)_{\neq}$

x c b a

*_____*_____*_____*

$(bx+c)_=$ $(b)_{\neq}$ $(a)_=$

x c b a

*_____*_____*_____*

$(c)_=$ $(b)_=$ $(a)_=$

3.4. Descomposición algebraica de Thomas

Ahora veremos nuestro algoritmo principal para sistemas algebraicos y sus subalgoritmos. Cada sistema es un par que consta de un sistema simple candidato y una cola de ecuaciones e inecuaciones sin procesar. En cada paso, el algoritmo elige un polinomio adecuado de la cola, lo pseudo-reduce y luego lo combina con el polinomio del sistema candidato simple que tiene la misma variable líder. En este proceso, también puede pasar que el algoritmo divida el sistema, es decir, se podrá añadir a la cola un nuevo polinomio como

una inecuación y por tanto tendríamos dos sistemas, un sistema agregando la inecuación en la cola y otro sistema agregando a la cola el mismo polinomio pero como una ecuación.

Definición 3.4.1: Diremos que un par de sistemas (S_T, S_Q) es presimple si cumple lo siguiente:

1) S_T es triangular, por lo tanto, $|S_{T_{x_i}}| \leq 1$, $\forall 1 \leq i \leq n$ y $S_T \cap \{C_=, C_{\neq} | C \in K\} = \emptyset$.

2) Cumple una propiedad de los coeficientes líderes modificada: $\forall i$, $1 \leq i \leq n$, $p \in S_{T_{x_i}}$ y $a \in Sol((S_T)_{<x_i}) \cup (S_Q)_{<x_i})$ entonces $\phi_a(init(p)) \neq 0$.

3) $\forall i$, $1 \leq i \leq n$, $\forall a \in Sol((S_T)_{<x_i}) \cup (S_Q)_{<x_i})$, $p \in S_{T_{x_i}}$ entonces $\phi_{<x_i,a}(p) \in \overline{K}[x_i]$ es libre de cuadrados.

Observación 3.4.2: Si $S_Q = \emptyset$, el sistema (S_T, S_Q) es presimple $\iff$ S_T es simple.

A partir de ahora, diremos que *prem* es un **algoritmo pseudo-residuo** y *pquo* el correspondiente **algoritmo pseudo-cociente**. Para ser precisos sean $p, q \in R$ con $ld(p) = ld(q) = x$, entonces,

$$m \cdot p = pquo(p, q, x) \cdot q + prem(p, q, x) \tag{8}$$

donde $deg_x(q) > deg_x(prem(p, q, x))$, $ld(m) < x$ y $m | init(q)^k$ para algún $k \in \mathbb{Z}_{\geq 0}$. Remarcar que pquo y prem no son únicos.

Veremos ahora un algoritmo para calcular prem y pquo.

Algoritmo 3.4.3: El algoritmo para encontrar el prem y pquo utilizando la división de polinomios es el siguiente:

Si p es cero, devolver $pquo = 0$ y $prem = 0$ ya que cualquier número dividido por cero es cero con un resto de cero.

Si q es igual a 1, devolver $pquo = p$ y $prem = 0$ ya que cualquier número dividido por uno es igual al número con un resto de cero.

Si $ld(p)$ es menor que $ld(q)$, devolver $pquo = 0$ y $prem = p$ ya que el cociente es cero y el resto es igual a p.

Sean $p = p_0 + ... + p_m x^m$ y $q = q_0 + ... q_n x^n$ los dos polinomios a dividir con $mdeg(p) = m$, con $p_m \neq 0$ y $mdeg(q) = n$, con $q_n \neq 0$.

Como para hacer la división de dos polinomios lo usual para poder resolverla sería dividir el coeficiente líder de p entre el polinomio líder de q, es decir, p_n/q_n, pero no siempre podremos hacer eso.

Por lo tanto la manera de actuar será la siguiente, teniendo en cuenta que hay otros algoritmos válidos que cumplen las especificaciones y que dan resultados diferentes.

Un algoritmo de pseudo-división:

Entrada: Un polinomio p y un polinomio $q \neq 0$, con $ld(p) = ld(q) = x$.

Salida: Un polinomio $pquo := c$, un polonimio $prem := r$, cumpliendo (8) y las respectivas condiciones.

$r := p$.

$c := 0$.

while $r \neq 0$ y $deg(r, x) \geq deg(q, x)$ **do**

$r := init\,(q) \cdot r - init(r) \cdot q \cdot x^{(deg(r,x)-deg(q,x))}$,

$c := init\,(q) \cdot c + init(r) \cdot x^{(deg(r,x)-deg(q,x))}$,

end while

En la primera iteración obtenemos:

$r := init\,(q) \cdot q - init(q) \cdot q \cdot x^{(deg(r,x)-deg(q,x))}$,

$c := init\,(q) \cdot 0 + init(r) \cdot x^{(deg(r,x)-deg(q,x))}$,

obtenemos $pquo \cdot q + prem = init(q) \cdot q$, se prueba que en cada paso se obtiene un $init(q)$ más, es decir, aplicando el algortimo m-n+1 veces obtenemos lo que queremos.

$init(q)^{(m-n+1)} \cdot p = q \cdot pquo + prem$.

Lema 3.4.4: Si en la ecuación (8) tenemos que $\phi_a(init(p)) \neq 0$ y $\phi_a(init(q)) \neq 0$ implica $\phi_{<x,a}(pquo(p, q, x)) \neq 0$ y $\phi_a(m) \neq 0$

Demostración: Supongamos que variable líder es $x = x_k$.

Sea $p = p_0(x_1, \ldots, x_{k-1}) + p_1(x_1, \ldots, x_{k-1})x_k + \ldots + p_m(x_1, \ldots, x_{k-1})x_k^m$, donde $init(p) = p_m(x_1, \ldots, x_{k-1}) \neq 0$.

Sea $q = q_0(x_1, \ldots, x_{k-1}) + q_1(x_1, \ldots, x_{k-1})x_k + \ldots + q_n(x_1, \ldots, x_{k-1})x_k^n$, donde $init(q) = q_n(x_1, \ldots, x_{k-1}) \neq 0$.

Sea $pquo = t_0(x_1, \ldots, x_{k-1}) + t_1(x_1, \ldots, x_{k-1})x_k + \ldots + t_l(x_1, \ldots, x_{k-1})x_k^l$, donde $init(pquo) = t_l(x_1, \ldots, x_{k-1})$.

Sea $prem = r_0(x_1, \ldots, x_{k-1}) + r_1(x_1, \ldots, x_{k-1})x_k + \ldots + r_s(x_1, \ldots, x_{k-1})x_k^s$, donde $init(prem) = r_s(x_1, \ldots, x_{k-1})$.

Tenemos la identidad,

$$m \cdot p = pquo(p, q, x) \cdot q + prem(p, q, x), \tag{9}$$

y como sabemos que $m|init(q)^k$, entonces tenemos que $q_n(x_1, \ldots, x_{k-1})^k = m \cdot h$, donde $m(x_1, \ldots, x_{k-1})$ y $h(x_1, \ldots, x_{k-1})$ son polinomios en las variables $(x_1, \ldots, x_{k-1})$.

Veamos que $m = n + l$.

Por (9) y dado que $s < n$, el grado en x_k de la parte derecha de la igualdad (9) es $n + l$ y es igual a m, así, $m = n + l$.

La fórmula $q_n(a_1, \ldots, a_{k-1})^k = m(a_1, \ldots, a_{k-1}) \cdot h(a_1, \ldots, a_{k-1})$ implica que como $q_n(a_1, \ldots, a_{k-1})^k \neq 0$, entonces $m(x_1, \ldots, x_{k-1}) \neq 0$ y $h(x_1, \ldots, x_{k-1}) \neq 0$.

Por tanto tenemos ya que como $\phi_a(m) = \phi_{<x,a}(m)$, ya que m sólo depende de $x_1, .., x_{k-1}$, que $\phi_a(m) \neq 0$.

Ahora sólo nos queda demostrar que $\phi_{<x,a}(pquo(p, q, x)) \neq 0$.

En la ecuación,

$$m \cdot p = pquo(p, q, x) \cdot q + prem(p, q, x),$$

aplicamos la función $\phi_{<x,a}$, y tenemos que

$$\phi_a(m) \cdot \phi_{<x,a}(p) = \phi_{<x,a}(pquo(p, q, x)) \cdot \phi_{<x,a}(q) + \phi_{<x,a}(prem(p, q, x)),$$

dónde observamos que $\phi_a(m) = \phi_{<x,a}(m)$, ya que m sólo depende de $x_1, .., x_{k-1}$.

Al tomar $\phi_{<x,a}$ obtenemos,

$m(a_1 \ldots a_{k-1}) \cdot \{p_0(a_1, \ldots, a_{k-1}) + \ldots + p_m(a_1, \ldots, a_{k-1})x_k^m\} =$

$\{t_0(a_1, \ldots, a_{k-1}) + \ldots + t_l(a_1, \ldots, a_{k-1})x_k^l\} \cdot \{q_0(a_1, \ldots, a_{k-1}) + \ldots + q_n(a_1, \ldots, a_{k-1})x_k^n\} +$

$\{r_0(a_1, \ldots, a_{k-1}) + \ldots + r_s(a_1, \ldots, a_{k-1})x_k^s\}$.

Donde sabemos que $m(a_1 \ldots a_{k-1}) \neq 0$ y $p_m(a_1, \ldots, a_{k-1}) \neq 0$, por tanto la parte izquierda de la ecuación es distinta de 0 y de grado m en x_k. Entonces la parte derecha tiene grado $m = n+l$, y como $s < n$, necesariamente $t_l(a_1, \ldots, a_{k-1}) \neq 0$, es decir, $\phi_{<x,a}(pquo(p, qx)) \neq 0$. □

El siguiente algoritmo Reduce lo que hace es reducir un polinomio de entrada p módulo S, es decir, utiliza las ecuaciones e inecucaciones del sistema S para obtener un polinomio de salida q con las mismas características que el polinomio p pero reduciendo incógnitas:

Algoritmo 3.4.5(Reduce o Reducir módulo S).

Entrada: Un sistema presimple $S = (S_T, S_Q)$, un polinomio $p \in R$.

Salida: Un polinomio q con $\phi_a(p) = 0$ si y solo si $\phi_a(q) = 0$ para cada $a \in Sol(S)$.

Algoritmo:

1: $x := ld(p)$; $q := p$;

2: ***while*** $x > 1$ y $(S_T)_x$ is an equation y $mdeg(q) \geq mdeg((S_T)_x)$ ***do***

3: $\quad q := prem(q, (S_T)_x, x)$;

4: $\quad x := ld(q)$;

5: ***end while***

6: ***if*** $x > 1$ y $Reduce(S, init(q)) = 0$ ***then***

7: $\quad$ ***return*** $Reduce(S, q - init(q)x^{mdeg(q)})$

8: ***else***

9: $\quad$ ***return*** q

10: ***end if***

A continuación veremos unas observaciones que nos ayudarán a entender el algoritmo y servirán de apoyo para la demostración.

Observación 3.4.6: Suponiendo que el algoritmo es correcto, la condición $Reduce(S, p) = 0$ implica que $\{p = 0\} \cap Sol(S) = Sol(S) \iff Sol(S) \subseteq \{p = 0\}$. El recíproco no se cumple en general.

Demostración de la corrección del algoritmo 3.4.5:

Lema 3.4.7: En primer lugar demostraremos que el polinomio q en la salida del primer bucle while es de la forma:

$$q = m \cdot p - \sum_{y \leq ld(p)} c_y \cdot (S_T)_y$$

con $c_y \in K[x|x \leq y]$. Además si p cumple las condiciones del bucle while se tiene que $(ld(q), mdeg(q)) < (ld(p), mdeg(p))$ y si no lo cumple $p = q$.

Demostración: Primero vamos a demostrar que el bucle while es finito.

Sean q_k y y_k los valores a la entrada de la $k-$ésima iteración del bucle while de las variables q y x respectivamente. En particular si se dan las condiciones para entrar en el bucle while, se tiene que, $q_1 = p$, $y_1 = ld(p)$. Tambien sabemos que, $q_{k+1} = prem(q_k, (S_T)_{y_k}, y_k)$, $y_{k+1} = ld(q_{k+1})$ y por lo tanto, $m_k \cdot q_k = pquo(q_k, (S_T)_{y_k}, y_k) \cdot (S_T)_{y_k} + q_{k+1}$.

Dónde tenemos que en cada paso $y_{k+1} < y_k$, ya que si no fuera así, se saldría del bucle while, esto es una sucesión decreciente y por tanto es finita, por tanto hemos demostrado lo que queríamos, es decir, que el bucle while es finito.

Ahora una vez demostrado la primera parte de la demostración veremos por inducción que q es de la forma del lema 3.4.7 La inducción se realiza en el número k de pasos que se da en bucle while.

Para $k = 1$, tenemos que $q = 1 \cdot p$. Se cumple.

Supongamos que se cumple para $k - 1$ y lo probaremos para k.

En el paso $k - 1$ hemos supuesto que se cumple la hipótesis de inducción, por tanto tenemos que,

$$q_{k-1} = m \cdot p - \sum_{l=1}^{k-2} c_{y_l} \cdot (S_T)_{y_l}, \tag{10}$$

donde

$$m = \prod_{l=1}^{k-2} m_l. \tag{11}$$

Para k, tenemos que,

$$m_{k-1} \cdot q_{k-1} = (S_T)_{k-1} \cdot pquo(q_{k-1}, (S_T)_{y_{k-1}}, y_{k-1}) + prem(q_{k-1}, (S_T)_{y_{k-1}}, y_{k-1}), \tag{12}$$

donde $q_k = prem(q_{k-1}, (S_T)_{y_{k-1}}, y_{k-1})$.

Sustituyendo en (12), q_{k-1} tenemos que,

$$q_k = m_{k-1}[m \cdot p - \sum_{l=1}^{k-2} c_{y_l} \cdot (S_T)_{y_l}] - (S_T)_{k-1} \cdot c_{k-1}.$$

Agrupando términos llegamos a,

$$q_k = m \cdot p - \sum_{l=1}^{k-1} c_{y_l} \cdot (S_T)_{y_l},$$

donde

$$m = \prod_{l=1}^{k-1} m_l.$$

Sea q la salida del primer while, hemos demostrado que q es de la forma:

$$q = m \cdot p - \sum_{y \leq ld(p)} c_y \cdot (S_T)_y \,, \tag{13}$$

con

$$m = \prod_{k=1}^{N} m_k.$$

Veamos ahora que se cumplen las especificaciones del agoritmo, es decir, que $\forall a \in Sol(S)$, $\phi_a(p) = 0 \iff \phi_a(q) = 0$.

Observación 3.4.8: Veamos como se comporta el algortimo. Si $a \in Sol(S_T \cup S_Q)$ y $p \in S_{y_k}$ entonces como S es presimple implica que $\phi_a(init(S_T)_{y_k}) \neq 0$.

Además como $m_k | init((S_T)_{y_k})^l$, entonces $\phi_a(m_k) \neq 0$.

Nos referimos a la línea 3 del algoritmo en el primer paso del bucle while del paso $k-$ésimo.

Sabemos por la observación 3.4.8 que $m_k | init((S_T)_{y_k})$.

Sea $a \in Sol(S)$, como $(S_T)_{y_k}$ es una ecuación, entonces $\phi_a((S_T)_{y_k}) = 0$. Lo que implica que, usando la fórmula (13) $\phi_a(q) = \phi_a(m) \cdot \phi_a(p)$.

Como $a \in Sol(S)$, entonces $a \in Sol(S_{\leq y_k})$. Teniendo en cuenta que $a \in Sol(S_{\leq y_k})$ y que S es presimple entonces, $\phi_a(init(S_T)_{y_k}) \neq 0$.

Sabiendo que $\phi_a(init(S_T)_{y_k}) \neq 0$ y $m_k | init(S_{y_k})$, implica que $\phi_a(m_k) \neq 0$.

Por tanto, $\phi_a(m) = \prod \phi_a(m_k) \neq 0$.

Luego llegamos a la afirmación $\phi_a(p) = 0 \iff \phi_a(q) = 0$.□

La inducción que usaremos para demostrar lo que lo que sale en el segundo bucle cumple las especificaciones es la doble inducción sobre $(ld(p), mdeg(p))$, donde $ld(p) \in \{0, 1, 2, \ldots, n\}$ donde llamamos 0 a la constante, 1 a x_1, y n a x_n y $mdeg(p) \in \mathbb{N}$. Se puede realizar esta inducción ya que los dos son conjuntos están bien ordenados y su producto está bien ordenado.

Veremos que la salida en el segundo bucle cumple las especificaciones del algoritmo:

Supongamos que $ld(q) = x_k$, y sea

$$q = q_0(x_1, \ldots, x_{k-1}) + q_1(x_1, \ldots, x_{k-1})x_k + \ldots + q_l(x_1, \ldots, x_{k-1})x_k^l,$$

donde $init(q) = q_l(x_1, \ldots, x_{k-1})$.

Sea $\bar{q} = Reduce(S, init(q))$. En el caso de que $Reduce(S, init(q)) \neq 0$, tenemos que la salida del algoritmo es q que es la salida del primer bucle y cumple las especificaciones por el lema 3.4.7 y por tanto cumple las especificaciones del algoritmo 3.4.5.

Por hipótesis de inducción dado que por el lema 3.4.7, o bien $(ld(q), mdeg(q)) < (ld(p), mdeg(p))$, o bien $q = p$ y evidentemente cumple las especificaciones.

Supongamos ahora que $\bar{q} = \bar{0}$.

Como $(ld(init(q)), mdeg(init(q))) < (ld(q), mdeg(q)) \leq (ld(p), mdeg(p))$ por hipótesis de inducción $\bar{q}$ cumple con las especificaciones del algortimo y por lo tanto, $\forall a \in Sol(S), \phi_a(init(q)) = 0$. Ahora definimos $\tilde{q} = q - q_l(x_1, \ldots, x_{k-1})x_k^l$, observamos $mdeg(\tilde{q}) < mdeg(q)$.

Por hipótesis de inducción, podemos suponer que $\hat{q} = Reduce(S, \tilde{q})$ cumple con las especificaciones del algoritmo y entonces: $\forall a \in Sol(S), \phi_a(\tilde{q}) = 0 \iff \phi_a(\hat{q}) = 0$.

Entonces como $\tilde{q} = q - q_l(x_1, \ldots, x_{k-1})x_k^l$, tenemos $\phi_a(\tilde{q}) = \phi_a(q) - \phi_a(q_l)a_k^l$.

Sabemos por el lema 3.4.7 que $\forall a \in Sol(S)$, $\phi_a(p) = 0 \iff \phi_a(q) = 0$.

Como $\forall a \in Sol(S)$, $\phi_a(q_l) = 0$, así $\phi_a(\tilde{q}) = \phi_a(q)$.

Anteriormente hemos visto que $\forall a \in Sol(S)$, $\phi_a(\tilde{q}) = 0 \iff \phi_a(\hat{q}) = 0$, por lo tanto tenemos que $\forall a \in Sol(S)$, $\phi_a(p) = 0 \iff \phi_a(\hat{q}) = 0$. $\square$

Observamos que este algoritmo solo usa las ecuaciones de S_T, ni las inecuaciones de S_T, ni los elementos de S_Q.

Diremos que p ***se reduce*** a q ***módulo*** S si $Reduce(S, p) = q$,y que p es ***reducido módulo*** S si se reduce a si mismo.

Observación 3.4.9: Sea $q = Reduce(S, p) \neq 0$. Entonces se tienen las siguientes propiedades:

1. Si $(S_T)_{ld(q)}$ es una ecuación, entonces $mdeg(q) < mdeg((S_T)_{ld(q)})$. Si $mdeg(q) \geq mdeg((S_T)_{ld(q)})$ entonces q no puede ser salida directa del bucle while.

2. $Reduce(S, init(Reduce(S, p))) \neq 0$. Ya que si no fuera así, sabiendo que $Reduce(S, p) = q$, entonces tendríamos $Reduce(S, init(q)) = 0$, entonces al entrar en el segundo bucle no saldría $q = Reduce(S, p)$.

3. Se tiene que $ld(q) \leq ld(p)$ y si $ld(q) = ld(p)$, entonces $mdeg(q) \leq mdeg(p)$. Esto se obtiene de la relación $(ld(q), mdeg(q)) \leq (ld(p), mdeg(p))$.

Observación 3.4.10: Sean $p \in R$ y $x = ld(p)$.

1. Si $(S_Q)_{\leq x} = \emptyset$, entonces, $S_{\leq x} = (S_T)_{\leq x}$ es simple y además $Reduce(S, p) \neq 0$ implica que $\exists a \in Sol(S_{\leq x})$ tal que $\phi_a(p) \neq 0$.

2. Si $(S_Q)^{=}_{<x} = \emptyset$ y $Reduce(S, p) \neq 0$, entonces $Sol(S_{<x}) = \emptyset$ o $\exists a \in Sol(S_{<x} \cup \{(S_T)_x\})$ tal que $\phi_a(p) \neq 0$.

Ejemplo 3.4.11 (Sobre el algoritmo Reduce): Reduce $p := x^2 + y^2x + x + y$ módulo el sistema simple siguiente $S_T := \{S_x, S_y\}$.

x	y
$(S_T)_x = (yx^2 - 1)_=$	$(S_T)_y = (y^2 + 1)_=$

$p := x^2 + y^2x + x + y$ al multiplicar $y \cdot p - (S_T)_x$ obtenemos que $(y^3 + y)x + y^2 + 1 =: p_2$

De aquí sabemos que $init(p_2) = y^3 + y$, y además $init(p_2) - y \cdot (S_T)_y$ es 0.

Por otro lado al obtener $p_3 := y^2 + 1$, es claro que $p_3 - (S_T)_y = 0$.

Vamos a explicar los pasos seguidos en el anterior ejemplo.

En este ejemplo lo que queremos encontrar es un polinomio q tal que $Reduce(S_T, p) = q$, es decir, dado los datos enunciados utilizar el algoritmo reduce para encontrar este polinomio.

Sigamos el algoritmo paso a paso.

No utilizaremos el nombre de q hasta el final para evitar problemas con usar siempre el mismo nombre, por este motivo utilizaremos p_i en cada paso para evitar confusiones.

En el primer paso del algoritmo llamamos $q := p = x^2 + y^2x + x + y$, $ld(p_1) = x$.

Entramos en el primer bucle while ya que $x > 1$, $(S_T)_x$ es una ecuación y $mdeg(p_1) = 2 \geq mdeg((S_T)_x) = 2$, ahora tenemos que calcular $prem(p_1, (S_T)_x, x)$, que al multiplicar $y \cdot p_1$ y al restarlo de $(S_T)_x$, obtenemos que $prem(p_1, (S_T)_x, x) = p_2 = (y^3 + y)x + y^2 + 1$ que es lo que sería el nuevo polinomio y además $ld(p_2) = x$.

Al intentar volver a entrar al primer bucle while se sale, ya que $mdeg(p_2) = 1 < mdeg((S_T)_x) = 2$. Por lo tanto pasamos al siguiente bucle.

Necesitamos calcular $Reduce(S_T, init(p_2))$, donde $init(p_2) = (y^3 + y)$. Volvemos al bucle while, y tenemos que $y > 1$, $(S_T)_y$ es una ecuación y $mdeg(p_2) = 3 > mdeg((S_T)_y) = 2$, luego podemos entrar al bucle y calculamos $prem(p_2, (S_T)_y, y) = p_3 = 0$, que lo obtenemos al multiplicar $init(p_2) \cdot 1 - y \cdot (S_T)_y = 0$, por lo tanto $Reduce(S_T, init(p_2)) = 0$.

Al estar aqui tenemos que devolver $Reduce(S_T, p_2 - init(p_2) \cdot x^1)$.

Calculemos ahora $Reduce(S_T, y^2+1)$, donde llamaremos $p_3 = y^2+1$. Es claro que al entrar en el primer bucle while, $y > 1$, $(S_T)_y$ es una ecuación y $mdeg(p_3) = 2 \geq mdeg((S_T)_y) = 2$ entra en el bucle, y como $p_3 - (S_T)_y = 0$, entonces sale 0 del bucle while, y al final del algoritmo devuelve 0.

Por tanto hemos terminado el ejemplo con la solución $Reduce(S_T, p) = 0$.

3.5. Algoritmo de Thomas

El algoritmo de Thomas consiste en dado un conjunto de sistemas, ir eligiendo sistemas, que desaparecerá del conjunto de sistemas. Vamos a explicar un poco como funcionaría y a continuación veremos los subalgoritmos implementos en el algoritmo principal para obtener lo deseado. En primer lugar al elegir cada sistema, tendremos que mirar a la cola, si la cola esta vacío, sumaremos a la salida del algoritmo el sistema al resultado y volveremos a escoger otro sistema. En el caso de que la cola no esté vacía, escogemos una ecuación o inecuación de la cola y esa ecuación o inecuación desaparecerá de la cola. Antes de ver los siguientes pasos tendremos que comprobar varias cosas. La primera tenemos que verificar si la ecuación o inecuación son consistentes, es decir, si la parte izquierda de la ecuación es una constante distinta de cero o si la parte izquierda de la inecuación es cero, entonces serían inconsistentes y lo desechamos. Por otra parte si tenemos que la ecuación elegida al lado izquierdo es cero, o si la inecuación elegida es una constante distinta de cero, lo desechamos. Para seguir con el algoritmo sólo quedaría una opción, que las ecuaciones o inecuaciones elegidas sean polinomios no constantes. En lo que queda del algoritmo intervienen el algoritmo Reduce visto en la anterior subsección y los subalgortimos que veremos a continuación. El objetivo de este algoritmo es, que utilizando lo anterior y utilizando en cada caso difererente los algoritmos y subalgortimos, tener que todos los conjuntos son triangulares, y que las condiciones iniciales de las ecuaciones e inecuaciones de cada sistema no desaparezcan para ninguna solución del respectivo sistema.

Algoritmo 3.5.1 (Split):

Entrada: Un sistema S y un polinomio $p \in R$.

Salida: La decomposición disjunta $(S \cup \{p_{\neq}\}, S \cup \{p_{=}\})$ de S.

Algoritmo:

1: **return** $((S_T, S_Q \cup \{p_{\neq}\}), (S_T, S_Q \cup \{p_{=}\}))$

Observación 3.5.2: $Sol(S) = Sol(S \cup \{p_{\neq}\}) \cup Sol(S \cup \{p_{=}\})$.

Algoritmo 3.5.3 (Init-Split):

Entrada: Un sistema S, una ecuación o inecuación q con $ld(q) = x$.

Salida: Dos sistemas S_1 y S_2, donde $(S_1 \cup \{q\}, S_2)$ es una descomposición disjunta de $S \cup \{q\}$. Además, $\phi_a(init(q)) \neq 0\ \forall a \in Sol(S_1)$ y $\phi_a(init(q)) = 0\ \forall a \in Sol(S_2)$.

Algoritmo:

1: $(S_1, S_2) \leftarrow Split(S, init(q))$

2: $(S_2)_Q \leftarrow (S_2)_Q \cup \{q\}$

3: **return** (S_1, S_2)

Observación 3.5.4: $Sol(S) = Sol(S_1 \cup \{q\}) \cup Sol(S_2)$.

Para ver algoritmos de división adicionales necesitamos ver nuevas definiciones y conceptos como preparación.

Definición 3.5.5: Sean $p, q \in R$ con $ld(p) = ld(q) = x$, $deg_x(p) = d_p > deg_x(q) = d_q$. Denotamos por $PRS(p, q, x)$ la secuencia del resto polinomial subresultante de p y q con respecto a x y por $PRS_i(p, q, x)$, $i < d_q$ el polinomio regular de grado i en $PRS(p, q, x)$ si existe, y si no existe denotándolo 0. Además, $PRS_{d_p}(p, q, x) := p$, $PRS_{d_q}(p, q, x) := q$ y $PRS_i(p, q, x) := 0$, $d_q < i < d_p$.

Definamos $res_i(p, q, x) := init(PRS_i(p, q, x))$ para $0 < i < d_p$, $res_{d_p}(p, q, x) := 1$ y $res_0(p, q, x) := PRS_0(p, q, x)$. Notar que $res_0(p, q, x)$ es el resultante habitual.

En el capítulo 2 hemos visto esto pero con otras notación, donde $PRS_i = F_j$ tal que $deg(F_j) = i$ o bien 0 si $i \notin \{deg(F_1), deg(F_2), \cdots, deg(F_k)\}$. Por tanto $deg(mcd(p, q)) = deg(F_k) = min\{j / PRS_j \neq 0\}$.

Es importante las iniciales de los subresultantes porque lo vamos a utilizar para poner condiciones para poder determinar los grados de los mcd.

Definición 3.5.6: Sea S un sistema y $p_1, p_2 \in R$ con $ld(p_1) = ld(p_2) = x$. Si $|Sol(S_{<x})| > 0$, llamamos:

$i := min\{i \in \mathbb{Z}_{\geq 0} | \exists a \in Sol(S_{<x})$ tal que $deg_x(gcd(\phi_{<x,a}(p_1)), \phi_{<x,a}(p_2)) = i\}$

la **cardinalidad de fibra** de p_1 y p_2 con respecto a S.

Además, si $(S_Q)^{=}_{<x} = \emptyset$, tenemos:

$i' := min\{i \in \mathbb{Z}_{\geq 0} | Reduce(S_T, res_j(p_1, p_2, x)) = 0\ \forall j < i$ y $Reduce(S_T, res_i(p_1, p_2, x)) \neq 0\}$

es la **cardinalidad de cuasi-fibra** de p_1 y p_2 respecto a S.

Una descomposición disjunta (S_1, S_2) de S tal que:

1. $deg_x(mcd(\phi_{<x,a}(p_1), \phi_{<x,a}(p_2))) = i\ \forall a \in Sol((S_1)_{<x})$ y

2. $deg_x(mcd(\phi_{<x,a}(p_1), \phi_{<x,a}(p_2))) > i \ \forall a \in Sol((S_2)_{<x})$

se llama **i-ésima división de fibración** de p_1 y p_2 respecto a S. Un polinomio $r \in R$ con $ld(r) = x$ tal que $deg_x(r) = i$ y

$\phi_{<x,a}(r) \sim mcd(\phi_{<x,a}(p_1), \phi_{<x,a}(p_2)) \ \forall a \in Sol((S_1)_{<x})$

se llama **se llama i-ésimo máximo común divisor condicional** de p_1 y p_2 respecto a S. Por otro lado, $q \in R$ con $ld(q) = x$ y $deg_x(q) = deg_x(p_1) - i$ tal que

$$\phi_{<x,a}(q) \sim \frac{\phi_{<x,a}(p_1)}{mcd\left(\phi_{<x,a}(p_1), \phi_{<x,a}(p_2)\right)} \forall a \in Sol((S_1)_{<x})$$

se llama **cociente condicional** de p_1 por p_2 con respecto a S.

Al reemplazar $\phi_{<x,a}(p_2)$ en la anterior definición con $\frac{\partial}{\partial x}(\phi_{<x,\mathbf{a}}(p_1))$, obtenemos una **i-ésima división libre de cuadrados** y una **i-ésima parte condicional libre de cuadrados** de p_1 con respecto a S.

Ejemplo 3.5.7: Considere el sistema $S := \{(x^3 + y)_=\}$ y el polinomio $q := x^2 + x + y + 1$ con $y < x$. Calcular $res_0(S_x, q, x)$, $res_1(S_x, q, x)$ y $res_2(S_x, q, x)$. Antes de calcular hay que tener en cuenta la definición 3.5.5, para saber que $res_i = init(PRS_i)$. También para el calculo de PRS_i podemos ir al capítulo anterior y ver cómo se calculan. En este ejemplo tomaremos de ayuda Maple para obtener PRS y así calcular res_i.

Calculando con Maple obtenemos que $PRS_0(S_x, q, x) = y^3+7y^2+5y+1$, $PRS_1(S_x, q, x) = -xy+2y+1$ y $PRS_2(S_x, q, x) = x^2+x+y+1$. Por tanto como los res_i son los init de PRS_i, tenemos que $res_0(S_x, q, x) = y^3 + 7y^2 + 5y + 1$, $res_1(S_x, q, x) = -y$ y $res_2(S_x, q, x) = 1$. Ya que existe y_0 tal que $res_0(S_x, q, x) \neq 0$, luego esto significa que S_x y q no tienen factor común, por tanto $mcd(S_x, q) = 1$.

La cardinalidad de la fibra de S_x y q con respecto a S es 0.

La división cero de la fibración viene dada por $S_1 := S \cup \{(res_0(S_x, q, x))_{\neq}\}$ y $S_2 := S \cup \{(res_0(S_x, q, x))_=\}$.

La cardinalidad de la fibra con respecto de S_2 es 1.

La primera división de fibración viene dada por $S_{2,1} := S_2 \cup \{(-y)_{\neq}\}$ y $S_{2,2} := S_2 \cup \{(-y)_=\}$. Notar que en este caso $Sol(S_{2,1}) = Sol(S_2)$ y $Sol(S_{2,2}) = \emptyset$.

El cociente condicional cero de S_x y q es S_x.

El primer gcd condicional y el primer cociente condicional son $-yx + 2y + 1$ y $y^2x^2 + (2y^2 + y)x + 4y^2 + 4y + 1$, respectivamente.

Generalmente es muy costoso y díficil calcular la cardinalidad de la fibra directamente. Pero en el caso de que la cardinalidad de la cuasi-fibra sea estrictamente más pequeña que la cardinalidad de la fibra, la división de fibración nos llevará a un sistema inconsistente, y uno donde la cardinalidad de la cuasi-fibra se incrementa.

Lema 3.5.8: Sea $|Sol(S_{<x})| > 0$ y $(S_Q)^{=}_{<x} = \emptyset$. Para p_1 y p_2 como en la definición 3.5.6 con $\phi_a(init(p_1)) \neq 0 \ \forall a \in Sol(S_{<x})$ y $mdeg(p_1) > mdeg(p_2)$, sea i la cardinalidad de la fibra de p_1 y p_2 con respecto a S e i' la cardinalidad de la cuasi-fibra. Luego:

$$i' \leq i$$

donde la igualdad se da si y sólo si $|Sol(S_{<x} \cup \{res_{i'}(p_1, p_2, x)_{\neq}\})| > 0$.

Demostración: Sea $a \in Sol(S_{<x})$, $mdeg(p_1) > mdeg(p_2)$, $d_{p_1} := deg_x(p_1) = deg_x(\phi_{<x,a}(p_1))$, por hipótesis, ya que hemos supuesto que $\phi_a(init(p_1)) \neq 0$, $d_{p_2} := deg_x(p_2)$ y $d_{p_2,a} := deg_x(\phi_{<x,a}(p_2))$.

Si $i < max(d_{p_1}, d_{p_2,a}) - 1 = d_{p_1} - 1$ esto implica que, usando el corolario 2.6.7, es decir, si $max(d_{p_1}, d_{p_2,a}) = d_{p_1}$, que es un caso particular del corolario 2.6.7.

$$\phi_{<x,a}(PRS_i(p_1, p_2, x)) \sim PRS_i(\phi_{<x,a}(p_1), \phi_{<x,a}(p_2), x) \tag{2}$$

y

$$\phi_a(res_i(p_1, p_2, x)) = 0 \iff res_i(\phi_{<x,a}(p_1), \phi_{<x,a}(p_2), x) = 0 \tag{3}$$

Para llegar a estas igualdades hay que tener en cuenta que $\phi_{<x,a}(init(PRS_i(p_1, p_2, x))) = \phi_a(init(PRS_i(p_1, p_2, x)))$ y que $init(PRS_i(p_1, p_2, x)) = res_i(p_1, p_2, x)$.

Las condiciones (2) y (3) por definición también se cumplen para los casos triviales $d_{p_2} \leq i \leq d_{p_1}$, ya que en el corolario se cumple para $0 \leq i < max(d_{p1}, d_{p2}, x)$.

Para todos índices $j < i'$, como $a \in Sol((S_T)_{<x})$, a se puede extender a una solución a' de S_T por ser S_T triangular, y por la observación 3.4.8, ya que tenemos que $Reduce(S_T, res_j(p_1, p_2, x)) = 0$, $\phi_{a'}(res_i(p_1, p_2, x) = 0)$, pues $ld(res_i(p_1, p_2, x)) < x$.

Por (3) se sigue que $res_j(\phi_{<x,a}(p_1), \phi_{<x,a}(p_2), x) = 0$. Aplicamos el teorema 2.7.9 sucesivamente y obtenemos $PRS_j(\phi_{<x,a}(p_1), \phi_{<x,a}(p_2), x) = 0$. Por lo tanto, se tiene,

$$deg_x(mcd(\phi_{<x,a}(p_1), \phi_{<x,a}(p_2))) \geq i' \tag{4}$$

Esto implica que $i' \leq i$.

La igualdad en (4) se da si y sólo si existe $a \in Sol(S_{<x})$ tal que $\phi_a(res_{i'}(p_1, p_2, x)) \neq 0$.

Por lo tanto, $i = i'$ si y sólo si $|Sol(S_{<x} \cup \{res_{i'}(p_1, p_2, x)_{\neq}\})| > 0$. □

Nota: En el caso del anterior teorema no se puede aplicar si dos polinomios tienen el mismo grado. Para poder aplicarlo en este caso (mismo grado) veremos que en el siguiente

resultado que las iniciales no deben desaparecer.

Corolario 3.5.9: Sea $|Sol(S_{<x})| > 0$ y $(S_Q)^{=}_{<x} = \emptyset$. Para dos polinomios p_1 y p_2 como en la definición 3.5.6 con $\phi_a(init(p_1)) \neq 0$ y $\phi_a(init(p_2)) \neq 0$ $\forall a \in Sol(S_{<x})$, sea i la cardinalidad de la fibra de p_1 y p_2 con respecto a S e i' la cardinalidad de la cuasi-fibra de p_1 y $prem(p_2, p_1, x)$ con respecto a S. Luego:

$$i' \leq i$$

con la igualdad si y sólo si $|Sol(S_{<x} \cup \{res_{i'}(p_1, prem(p_2, p_1, x), x)_{\neq}\})| > 0$.

Demostración: Sea $a \in Sol(S_{<x})$. Por las hipótesis sobre las iniciales , por el corolario 2.3.8 implica que $\phi_{<x,a}(prem(p_2, p_1, x)) = prem(\phi_{<x,a}(p_2), \phi_{<x,a}(p_1), x)$.

Los polinomios univariantes $\phi_{<x,a}(p_1)$ y $\phi_{<x,a}(p_2)$ tienen el mismo mcd como $\phi_{<x,a}(p_1)$ y $prem(\phi_{<x,a}(p_2), \phi_{<x,a}(p_1), x)$.

Ahora podemos reemplazar p_2 con $prem(p_2, p_1, x)$ en el lema anterior. □

El siguiente algoritmo calcula la cardinalidad de las cuasi-fibras de dos polinomios respecto de un sistema S.

Algoritmo 3.5.10 (ResSplit):

Entrada: Un sistema S con $(S_Q)^{=}_{<x} = \emptyset$, dos polinomios $p, q \in R$ con $ld(p) = ld(q) = x$, $mdeg(p) > mdeg(q)$ y $\phi_a(init(p)) \neq 0$ $\forall a \in Sol(S_{<x})$.

Salida: La cardinalidad de la cuasi-fibra i de p y q con respecto a S y una división de la fibra i-ésima (S_1, S_2) de p y q con respecto a S.

Algoritmo:

1: i $\leftarrow min\{i \in \mathbb{Z}_{\geq 0} \ |Reduce(S_T, res_j(p, q, x)) = 0 \ \forall j < i$ and $Reduce(S_T, res_i(p, q, x)) \neq 0\}$

2: **return** $(i, S_1, S_2) := (i, split(S, res_i(p, q, x)))$

Demostración: Supongamos que $|Sol((S_l)_{<x})| > 0$, $l = 1, 2$, del caso contrario el enunciado es trivial.

El polinomio $g := PRS_i(\phi_{<x,a}(p), \phi_{<x,a}(q), x)$ no es identicamente 0 ya que, $(init(g))_{\neq} = (res_i(p, q, x))_{\neq} \in (S_1)_Q$, usando las condiciones (2) y (3) de la demostración del lema 3.5.8.

El grado de g es i y $g \sim mcd(\phi_{<x,a}(p), \phi_{<x,a}(q))$, como hemos probado en el lema 3.5.8.

Sea $a \in Sol((S_2)_{<x})$. Por el teorema 2.7.9 y que $(init(g))_{=} = (res_i(p, q, x))_{=} \in (S_2)_Q$ implica que $g \equiv 0$, por las condiciones mencionadas anteriormente.

Por lo tanto, $deg_x(gcd(\phi_{<x,a}(p), \phi_{<x,a}(q))) > i$. $\square$

Aplicamos la cardinalidad de la fibra y la división de fibración para calcular el máximo común divisor entre un polinomio de S_T y otro polinomio.

Algoritmo 3.5.11 (ResSplitGCD):

Entrada: Un sistema S con $(S_Q)^{=}_{<x} = \emptyset$, donde $(S_T)_x$ es una ecuación, y una ecuación $q_=$ con $ld(q) = x$. Además, $mdeg(q) < mdeg((S_T)_x)$.

Salida: Dos sistemas S_1 y S_2 y una ecuación $\tilde{q}_=$ tal que:

a) S_2=$\tilde{S}_2 \cup \{q\}$ donde (S_1,$\tilde{S}_2$) es una i-ésima fibración de $(S_T)_x$ y q con respecto a S.

b) $\tilde{q}$ es una i-ésimo máximo común divisor condicional de $(S_T)_x$ y q con respecto a S.

donde i es el cardinal de la cuasi-fibra de p y q con respecto a S.

Algoritmo:

1: $(i, S_1, S_2) \leftarrow ResSplit(S, (S_T)_x, q)$

2: $(S_2)_Q \leftarrow (S_2)_Q \cup \{q\}$

3: **return** $S_1, S_2, PRS_i((S_T)_x, q, x)_=$

Demostración:

La propiedad a) se sigue por el algoritmo 3.5.9.

La propiedad b) se obtiene por la demostración del algoritmo 3.5.9.

Nota: Notar que en este caso es imprescindible que $i > 0$, ya que $i = 0$ produciría una inconsistencia, ya que no tendría solución.

El algoritmo que veremos a continuación es similiar al anterior,pero en lugar de devolver el máximo común divisor, devuelve el primer polinomio de entrada dividido entre el máximo común divisor.

Algoritmo 3.5.12 (ResSplitDivide):

Entrada: Un sistema S con $(S_Q)^{=}_{<x} = \emptyset$ y dos polinomios p,q con $ld(p) = ld(q) = x$ y $\phi_a(init(p)) \neq 0$ $\forall a \in Sol(S_{<x})$. Además, si $mdeg(p) \leq mdeg(q)$ tenemos $\phi_a(init(q)) \leq 0$.

Salida: Dos sistemas S_1 y S_2 y un polinomio $\tilde{p}$ tal que:

a) S_2=$\tilde{S}_2 \cup \{q\}$ donde (S_1,$\tilde{S}_2$) es una i-ésima fibración de p y q' con respecto a S.

b) $\tilde{p}$ es una i-ésimo cociente condicional de p con respecto a S.

donde i es la cardinalidad de la cuasi-fibra de p y q con respecto a S, con $q' = q$ para $mdeg(p) > mdeg(q)$ y $q' = prem(q, p, x)$ en otro caso.

Algoritmo:

1: **if** $mdeg(p) \leq mdeg(q)$ **then**

2: **return** ResSplitDivide(S,p,prem(q,p,x))

3: **else**

4: $(i, S_1, S_2) \leftarrow ResSplit(S, p, q)$

5: **if** $i > 0$ **then**

6: $\tilde{\text{p}} \leftarrow pquo(p, PRS_i(p, prem(q, p, x), x), x)$

7: **else**

8: $\tilde{\text{p}} \leftarrow$ p

9: **end if**

10: $(S_2)_Q \leftarrow (S_2)_Q \cup \{q\}$

11: **return** S_1,S_2,$\tilde{\text{p}}$

12: **end if**

Demostración: De acuerdo con el corolario 3.5.8, podemos suponer sin pérdida de generalidad que $mdeg(p) > mdeg(q)$.

La propiedad a) se sigue del algoritmo 3.5.9.

$\forall a \in Sol(S_1)$, se tiene lo siguiente: Si $i = 0$, tenemos $deg_x(mcd(\phi_{<x,a}(p), \phi_{<x,a}(q'))) = 0$ y por tanto $\phi_{<x,a}(p)$ no comparte raíces con $\phi_{<x,a}(q')$.

Ahora sea $i > 0$. Por la fórmula $m \cdot p = pquo(p, q, x) \cdot q + prem(p, q, x)$ tenemos que:

$$m \cdot p = \tilde{\text{p}} \cdot PRS_i(p, q', x) + prem(p, PRS_i(p, q', x), x).$$

De acuerdo con el corolario 2.3.9 y

$\phi_{<x,a}(PRS_i(p_1, p_2, x)) \sim PRS_i(\phi_{<x,a}(p_1), \phi_{<x,a}(p_2), x)$

$\phi_a(res_i(p_1, p_2, x)) = 0 \iff res_i(\phi_{<x,a}(p_1), \phi_{<x,a}(p_2), x) = 0$

existe k_1 y $k_2 \in K\backslash\{0\}$ tal que

$\underbrace{\phi_a(m)}_{\neq 0} \cdot \phi_{<x,a}(p) = \phi_{<x,a}(\tilde{\text{p}}) \cdot \phi_{<x,a}(PRS_i(p, q, x)) + \phi_{<x,a}(prem(p, PRS_i(p, q, x), x))$

$$= \phi_{<x,a}(\tilde{\mathrm{p}}) \cdot k_1 PRS_i(\phi_{<x,a}(p), \phi_{<x,a}(q), x)$$
$$+k_2 prem(\phi_{<x,a}(p), \underbrace{PRS_i(\phi_{<x,a}(p), \phi_{<x,a}(q), x)}_{divide\phi_{<x,a}(p)}, x)$$

$= \phi_{<x,a}(\tilde{\mathrm{p}}) \cdot k_1 mcd(\phi_{<x,a}(p), \phi_{<x,a}(q)) + 0$ Luego, obtenemos la condición b) por

$$\phi_{<x,a}(\tilde{p}) \sim \frac{\phi_{<x,a}(p)}{mcd\left(\phi_{<x,a}(p), \phi_{<x,a}(q)\right)}$$

y $deg_x(\phi_{<x,a}(\tilde{p})) = deg_x(\phi_{<x,a}(p)) - deg_x(mcd(\phi_{<x,a}(p), \phi_{<x,a}(q))) = deg_x(p) - i$.

Aplicando el último algoritmo a p y la derivada de p respecto de su variable líder construye un algoritmo para hacer p libre de cuadrados. Lo presentaremos separado para entender mejor el algoritmo.

Algoritmo 3.5.13 (ResSplitSquareFree):

Entrada: Un sistema S con $(S_Q)^{=}_{<x} = \emptyset$ y un polinomio p con $ld(p) = x$ y $\phi_a(init(p)) \neq 0$ $\forall a \in Sol(S_{<x})$.

Salida: Dos sistemas S_1 y S_2 y un polinomio r tal que:

a) $S_2 = \tilde{\mathrm{S}}_2 \cup \{p\}$ donde (S_1,$\tilde{\mathrm{S}}_2$) es una i-ésima división libre de cuadrados de p con respecto a S.

b) r es una i-ésima parte libre de cuadrados condicional de p con respecto a S.

donde i es la cardinalidad de la cuasi-fibra de p y de $\frac{\partial}{\partial x}p$ con respecto a S.

Algoritmo:

1: $(i, S_1, S_2) \leftarrow ResSplit(S, p, \frac{\partial}{\partial x}p)$

2: **if** $i > 0$ **then**

3: $\quad r \leftarrow pquo(p, PRS_i(p, \frac{\partial}{\partial x}p, x), x)$

4: **else**

5: $\quad r \leftarrow p$

6: **end if**

7: $(S_2)_Q \leftarrow (S_2)_Q \cup \{p\}$

8: **return** S_1, S_2, r

Demostración: Ya que $\phi_{<x,a}(\frac{\partial}{\partial x}p) = \frac{\partial}{\partial x}\phi_{<x,a}(p)$, es una i-ésima división libre de cuadrados de p es una i-ésima división de fibración de p y $\frac{\partial}{\partial x}p$. El resto de la prueba se sigue por la demostración del algortimo 3.5.13.

En todos los algoritmos basados en ResSplit una condición necesaria es que $(S_Q)^{=}_{<x} = \emptyset$. Con esto aseguramos que con todas las ecuaciones con líder menor que x se pueden utilizar para la reducción módulo S_T. Esta restricción obliga a considerar un orden particular en el que tratar los polinomios en el algoritmo principal.

Definición 3.5.14 (Escoger): Sea $\mathbb{P}_{finito}(M)$ sea el conjunto de todos los subconjuntos finitos de un conjunto M. Una estrategia de selección es una aplicación:

Escoger:$\mathbb{P}_{finito}(\{p_=, p_{\neq}|p \in R\}) \to \{p_=, p_{\neq}|p \in R\}$:

$$Q \mapsto q \in Q$$

con las siguientes propiedades:

1. Si $Reducir(Q) = q_=$ es una ecuación, entonces $Q^{=}_{<ld(q)} = \emptyset$.

2. Si $Reducir(Q) = q_{\neq}$ es una inecuación, entonces $Q^{=}_{\leq ld(q)} = \emptyset$.

Demostramos que estas condiciones son necesarias para la terminación de nuestro enfoque, dando un ejemplo donde las incumplimos.

Ejemplo 3.5.15: Consideramos $R := K[a, x]$ con $a < x$ y el sistema S con $S_T := \emptyset$ y $S_Q := \{(x^2 - a)_=\}$. Para introducir $(x^2 - a)_=$ en S_T, necesitamos aplicar el algoritmo ResSplitSquareFree:

Calculamos $res_0(x^2 - a, 2x, x) = -4a$, $res_1(x^2 - a, 2x, x) = 2$ y $res_2(x^2 - a, 2x, x) = 1$ con la definición 3.5.5 La cardinalidad de la cuasi-fibra es 0 y obtenemos dos nuevos sistemas S_1 y S_2 con:

$$(S_1)_T = \{(x^2 - a)_=\},\ (S_1)_Q = \{(-4a)_{\neq}\} \text{ y } (S_2) = \emptyset,\ (S_2)_Q = \{(x^2 - a)_=, (-4a)_=\}.$$

Veremos ahora qué pasa con S_2: Si escogemos $(x^2 - a)_=$ como nueva ecuación para ser tratada, por incumplimiento de las condiciones en la definición 3.5.14, el algoritmo ResSplitSquareFree dividirá S_2 en $S_{2,1}$, $S_{2,2}$ con,

$$(S_{2,1})_T = \{(x^2 - a)_=\}, (S_{2,1})_Q = \{(-4a)_{\neq}, (-4a)_=\}$$

y

$$(S_{2,2})_T = \emptyset, (S_{2,2})_Q = \{(x^2 - a)_=, (-4a)_=\}.$$

Como $S_2 = S_{2,2}$, esto lleva a un bucle infinito.

El siguiente algoritmo trivial inserta una nueva ecuación en S_T.

Algoritmo 3.5.16 (Insertar una ecuación):

Entrada: Un sistema S y una ecuación $r_=$ con $ld(r) = x$ satisfaciendo $\phi_a(init(r)) \neq 0$ y $\phi_{<x,a}(r)$ es libre de cuadrados $\forall a \in Sol(S_{<x})$.

Salida: Un sistema S donde $r_=$ se inserta en S_T.

Algoritmo:

1: **if** $(S_T)_x$ es no vacío **then**

2: $\quad S_T \leftarrow (S_T \backslash \{(S_T)_x\})$

3: **end if**

4: $S_T \leftarrow S_T \cup \{r_=\}$

5: **return** S

Ahora presentamos técnicamente el algoritmo principal, pero sin detallar la demostración de su corrección y su finalización debido a que son muy técnicas y bastante largas, sin embargo ilustro el algoritmo con un ejemplo. También después del algoritmo incluyo un diagrama del flujo del algoritmo para una mejor comprensión. La estructura general es la siguiente: En cada iteración, un sistema S es seleccionado de una lista P de sistemas inacabados. Una ecuación o inecuación se elige de la cola S_Q. Entonces q se reduce módulo S_T y incorporado al sistema simple candidato S_T con los algortimos de división vistos anteriormente. Al hacerlo, el algoritmo puede agregar nuevos sistemas S_i a P. En cuanto el algoritmo produce un sistema que contiene una ecuación $c_=$ para $c \in K\ \{0\}$ o la inecuación $0_{\neq}$ se descarta.

Algoritmo 3.5.17 (Descomposición):

Entrada: Un sistema S' con $(S')_T = \emptyset$.

Salida: Una descomposición de Thomas de S'.

Algoritmo:

1: $P \leftarrow \{S'\}; Result \leftarrow \emptyset$

2: **while** $|P| > 0$ **do**

3: $\quad$ Escoger $S \in P; P \leftarrow P \backslash \{S\}$

4: $\quad$ **if** $|S_Q| = 0$ **then**

5: $\quad\quad Result \leftarrow Result \cup \{S\}$

6: **else**

7: $q \leftarrow Select(S_Q); S_Q \leftarrow S_Q \backslash \{q\}$

8: $q \leftarrow Reduce(q, S_T); x \leftarrow ld(q)$

9: **if** $q \notin \{0_{\neq}, c_{=} | c \in K \backslash \{0\}\}$ **then**

10: **if** $x \neq 1$ **then**

11: **if** q es una ecuación **then**

12: **if** $(S_T)_x$ es una ecuación **then**

13: **if** $Reduce(S_T, res_0((S_T)_x, q, x) = 0$ **then**

14: $(S, S_1, p) \leftarrow ResSplitGCD(S, q); P \leftarrow P \cup \{S_1\}$

15: $S \leftarrow InsertEquation(S, p_{=})$

16: **else**

17: $S_Q \leftarrow S_Q \cup \{q_{=}, res_0((S_T)_x, q, x)_{=}\}$

18: **end if**

19: **else**

20: **if** $(S_T)_x$ es una inecuación **then**

21: $S_Q \leftarrow S_Q \cup \{(S_T)_x\}; S_T \backslash \{(S_T)_x\}$

22: **end if**

23: $(S, S_2) \leftarrow InitSplit(S, q); P \leftarrow P \cup \{S_2\}$

24: $(S, S_3, p) \leftarrow ResSplitSquareFree(S, q); P \leftarrow P \cup \{S_3\}$

25: $S \leftarrow InsertEquation(S, p_{=})$

26: **end if**

27: **else if** q es una inecuación **then**

28: **if** $(S_T)_x$ es una ecuación **then**

29: $(S, S_4, p) \leftarrow ResSplitDivide(S, (S_T)_x, q); P \leftarrow \{S_4\}$

30: $S \leftarrow InsertEquation(S, p_{=})$

31: **else**

32: $(S, S_5) \leftarrow InitSplit(S, q); P \leftarrow P \cup \{S_5\}$

33: $(S, S_6, p) \leftarrow ResSplitSquareFree(S, q); P \leftarrow P \cup \{S_6\}$

34: **if** $(S_T)_x$ es una inecuación **then**

35: $(S, S_7, r) \leftarrow ResSplitDivide(S, (S_T)_x, p); P \leftarrow P \cup \{S_7\}$

36: $(S_T)_x \leftarrow (r \cdot p)_{\neq}$

37: **else if** $(S_T)_x$ es vacío **then**

38: $(S_T)_x \leftarrow p_{\neq}$

39: **end if**

40: **end if**

41: **end if**

42: **end if**

43: $P \leftarrow P \cup \{S\}$

44: **end if**

45: **end if**

46: **end while**

47: **return** Result

Demostraremos el algoritmo con un ejemplo simple. En cada paso, se omitiremos los sistemas inconsistentes que aparezcan.

Ejemplo 3.5.18: Sea $S = (S_T, S_Q) := (\emptyset, \{(x^2+x+1)_=, (x+a)_{\neq}\})$ con $a < x$. Escogemos $q := (x^2 + x + 1)_=$(Línea 7,8). Como $init(q) = 1$ y $res_0(q, \frac{\partial}{\partial x}p, x) = 1$(Línea 10) el sistema original S es reemplazado por $(\{(x^2 + x + 1)_=\}, \{(x + a)_{\neq}\})$ (Línea 43).

Ahora, $q := (x + a)_{\neq}$(Línea 7,8) es seleccionada y $ResSplitDivide(S, (S_T)_x, q)$ calcula $res_0((S_T)_x, q, x) = prem((S_T)_x, q, x) = a^2 - a + 1$, $res_1((S_T)_x, q, x) = init(q) = 1$, y $res_2((S_T)_x, q, x) = 1$. Como S_T no contiene ninguna ecuación del líder a(Línea 28), ninguno de esos polinomios pueden ser reducidos. Luego, podemos descomponer S en (Líneas 32-33):

$$S := (\underbrace{\{(x^2 + x + 1)_=, (a^2 - a + 1)_{\neq}\}}_{=S_T}, \underbrace{\{\}}_{=S_Q}),$$

que es simple y,

$$S_1 := (\underbrace{\{(x^2 + x + 1)_=\}}_{=(S_1)_T}, \underbrace{\{(x + a)_{\neq}, (a^2 - a + 1)_=\}}_{=(S_1)Q}).$$

Reemplazamos S_1 por

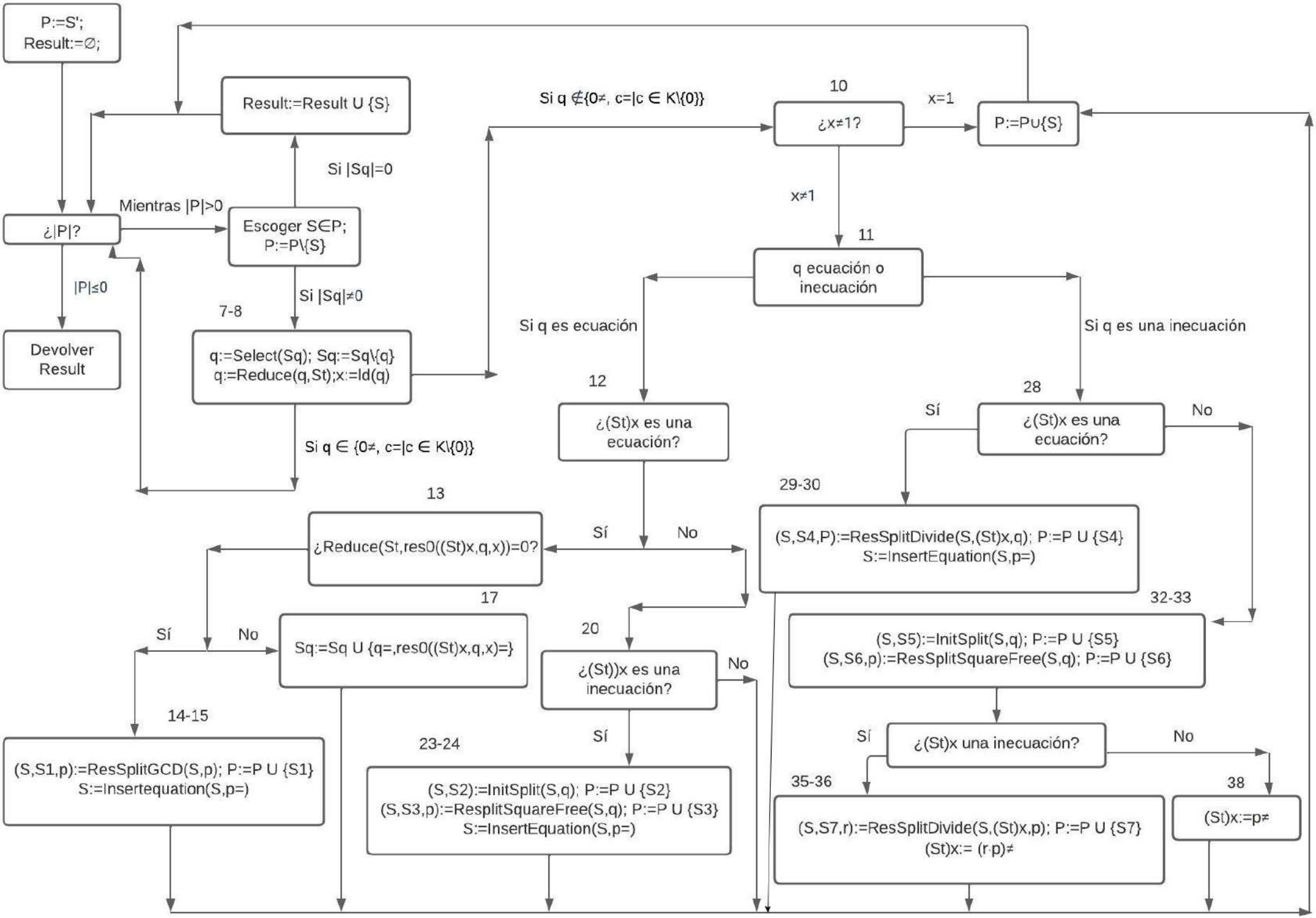
$$S_1 := (\{(x^2+x+1)_=, (a^2-a+1)_=\}, \{(x+a)_\neq\})$$

y aplicando $ResSplitDivide(S_1, ((S_1)_T)$ a S_1 otra vez. En este caso, $Reduce((S_1)_T, a^2 - a + 1) = 0$ y S_1 es reemplazado por (Líneas 35,36)

$$S_1 := (\{ \underbrace{(x-a+1)_=}_{pquo(x^2+x+1,x+a,x)}, (a^2-a+1)_=\}, \{1_\neq\})$$

Finalmente, la descomposición de Thomas de S es:

$(\{(x^2+x+1)_=, (a^2-a+1)_\neq\}, \{(x-a+1)_=, (a^2-a+1)_=\})$.


P:=S';
Result:=∅;
Result:=Result U {S}
Si |Sq|=0
Mientras |P|>0
¿|P|?
Escoger S∈P;
P:=P\{S}
|P|≤0
Devolver Result
Si |Sq|≠0
7-8
q:=Select(Sq); Sq:=Sq\{q}
q:=Reduce(q,St);x:=ld(q)
Si q ∈ {0≠, c=|c ∈ K\{0}}
Si q ∉{0≠, c=|c ∈ K\{0}}
10
¿x≠1?
x=1
P:=P∪{S}
x≠1
11
q ecuación o inecuación
Si q es ecuación
Si q es una inecuación
12
¿(St)x es una ecuación?
28
Sí
No
¿(St)x es una ecuación?
13
¿Reduce(St,res0((St)x,q,x))=0?
Sí
No
29-30
(S,S4,P):=ResSplitDivide(S,(St)x,q); P:=P U {S4}
S:=InsertEquation(S,p=)
17
Sí
No
Sq:=Sq U {q=,res0((St)x,q,x)=}
20
¿(St))x es una inecuación?
No
Sí
32-33
(S,S5):=InitSplit(S,q); P:=P U {S5}
(S,S6,p):=ResSplitSquareFree(S,q); P:=P U {S6}
Sí
¿(St)x una inecuación?
No
14-15
(S,S1,p):=ResSplitGCD(S,p); P:=P U {S1}
S:=Insertequation(S,p=)
23-24
(S,S2):=InitSplit(S,q); P:=P U {S2}
(S,S3,p):=ResplitSquareFree(S,q); P:=P U {S3}
S:=InsertEquation(S,p=)
35-36
(S,S7,r):=ResSplitDivide(S,(St)x,p); P:=P U {S7}
(St)x:= (r·p)≠
38
(St)x:=p≠

4. Bibliografía

Referencias

[1] Winkler,F. *Polynomial Algorithms in Computer Algebra*, Springer, 1996.

[2] Mishra,B. *Algorithmic Algebra*, Springer-Verlag, 1993.

[3] Woody,H. *Polynomial Resultant*, 2016.

[4] Janson,S. *Resultant and Discriminant of Polynomial*, 2007.

[5] Gelfand,I, Kapranov,M and Zelevinsky,A. *Discriminants, Resultants and Multidimensional Determinants*, 1994.

[6] Bächler,T , Gerdt,V , Lange-Hegermann,M and Robertz,D. *Algorithmic Thomas decomposition of algebraic and differential systems*, 1994.

[7] Sturmfels,B and Zelevinsky,A. *Multigraded resultants of Sylvester type*, 1994.

[8] Robertz,D. *Formal Algorithmic Elimination for PDEs*, 2014.

[9] Brown,W,S ,Traub,J,F. *On Euclid's algorithm and the theory of subresultants*, 1971.

[10] Collins,G,E. *Subresultants and reduced polynomial remainder sequences*, 1966.