UNIVERSIDAD DE BUENOS AIRES
Facultad de Ciencias Exactas y Naturales
Departamento de Matemática

Métodos simbólicos para sistemas de ecuaciones e inecuaciones pfaffianas sobre $\mathbb{R}$

Tesis presentada para optar al título de Doctora de la Universidad de Buenos Aires en el área Ciencias Matemáticas

María Laura Barbagallo

Directora de tesis: Dra. Gabriela Jeronimo.
Director adjunto: Dr. Juan Sabia.
Consejero de estudios: Dr. Daniel Perrucci.

Fecha de defensa: 27 de noviembre de 2019.

Métodos simbólicos para sistemas de ecuaciones e inecuaciones pfaffianas sobre $\mathbb{R}$

Resumen

En esta tesis desarrollamos herramientas algebraicas y métodos algorítmicos simbólicos para problemas que involucran funciones pfaffianas de orden 1 en una variable, es decir, funciones del tipo $f(x) = F(x, \varphi(x))$, con $F \in \mathbb{Z}[X, Y]$, donde φ es una función fija que es solución de una ecuación diferencial de la forma $\varphi'(x) = \Phi(x, \varphi(x))$, con $\Phi \in \mathbb{Z}[X, Y]$.

En una primera instancia, presentamos un nuevo procedimiento simbólico para contar la cantidad exacta de ceros de estas funciones en intervalos. Este procedimiento está basado en la construcción de secuencias de Sturm para este tipo de funciones y requiere, como es usual en la bibliografía, de un oráculo para la determinación del signo de estas funciones en números reales algebraicos. Abordamos también el problema de decisión para fórmulas que involucran funciones del mismo tipo construidas a partir de una función φ fija. En este contexto, introducimos una noción de secuencia de Sturm generalizada y presentamos un nuevo procedimiento simbólico basado en la construcción de estas secuencias que resuelve el problema de decisión con complejidad calculable, asumiendo nuevamente la existencia de un oráculo para la determinación de signos.

Para la clase particular de los E-polinomios, es decir, funciones con $\varphi(x) = e^{h(x)}$, $h \in \mathbb{Z}[X]$, desarrollamos algoritmos que resuelven los problemas anteriores sin necesidad de recurrir a oráculos y estimamos explícitamente sus complejidades. A continuación, aplicamos el algoritmo de decisión diseñado para resolver un problema de decisión similar en el caso de E-polinomios multivariados. Además, en el contexto de una variable, damos una cota superior explícita para el valor absoluto de los ceros reales de un E-polinomio. Finalmente, introducimos la noción de codificación de Thom para ceros de E-polinomios y describimos un algoritmo para su construcción.

Palabras claves: Funciones Pfaffianas; secuencias de Sturm; problema de decisión; complejidad.

Symbolic methods for systems of Pfaffian equations and inequalities over $\mathbb{R}$

Abstract

In this thesis we develop algebraic tools and algorithmic symbolic methods to deal with problems involving univariate Pfaffian functions of order 1, that is, functions of the type $f(x) = F(x, \varphi(x))$, with $F \in \mathbb{Z}[X, Y]$, where φ is a fixed univariate function satisfying a differential equation $\varphi'(x) = \Phi(x, \varphi(x))$, for $\Phi \in \mathbb{Z}[X, Y]$.

First, we present a new symbolic procedure to count the exact number of zeros of a function of this type in a real interval. This procedure is based on the construction of Sturm sequences for functions of this class and relies, as it is usual in the literature, on an oracle for determining the signs of these functions in real algebraic numbers. We also address the decision problem for formulas involving functions of the same kind constructed from a fixed function φ. In this setting, we introduce the concept of a generalized Sturm sequence and we present a new symbolic procedure, based on an explicit construction of these sequences, that solves the decision problem with a computable complexity also assuming the existence of an oracle for sign determination.

For the particular class of E-polynomials, namely, functions with $\varphi(x) = e^{h(x)}$, $h \in \mathbb{Z}[x]$, we design oracle-free effective algorithms that solve the previous problems and we compute explicit estimates for their complexities. We apply the decision algorithm developed to solve a similar decision problem for E-polynomials in the multivariate setting. In addition, in the univariate context, we give an explicit upper bound for the absolute value of the real zeros of an E-polynomial. Finally, we introduce a notion of Thom encoding for zeros of an E-polynomial and describe an algorithm for their computation.

Keywords: Pfaffian functions; Sturm sequences; decision problem; complexity.

Agradecimientos

A Gabriela y a Juan, por haberme aceptado en el grupo, por su ayuda infinita, por mostrarme el mundo de la investigación, por compartir sus puntos de vista, por enseñarme, por haber respondido mis dudas y repetido las respuestas... ¡Por las correcciones! Por haber respetado mis complicados tiempos todos estos años.

A la FCEyN de la UBA, por formarme. Eternamente orgullosa de la educación pública y gratuita.

Al Conicet, sin el apoyo económico de sus becas en los primeros años se hubiese puesto difícil.

A la oficina 2103, donde todo arrancó.

A Marie Francoise, a Alicia y a Nino, por aceptar ser los jurados de esta tesis y dedicarle tiempo a estas páginas.

A las dos personas más importantes de mi vida, mis dos amores. A Elias, que estuvo casi desde el comienzo del doctorado, que aguantó malos y buenos momentos, que apoyó y acompañó, que siempre, siempre, alentó a seguir adelante. A Vicky, que llegó para cambiarlo todo ¡Para hacerme más felíz! Tus risas, mimos y abrazos me dieron energía.

¡A las niñeras! A las tías Andrea y Angie, al tío Leo, a las abuelas Ebe y Delia: Gracias por cuidar a Vicky y darme tiempo para estudiar y trabajar.

A las chicas de la facu... a mis amigas Georgi, Pau, Anita, Flopa, Carito, Jime. ¡Las quiero tanto! Gracias por todas las charlas, el apoyo, sugerencias y ayuda estos años.

A mis compañeros de trabajo de Exactas, del CBC, de UTDT, de UNSAM... por los ratos compartidos.

Índice general

Introducción

Las funciones Pfaffianas son funciones analíticas reales que satisfacen sistemas triangulares de ecuaciones diferenciales de primer orden con coeficientes polinomiales. Esta clase de funciones, introducida por Khovanskii a fines de los 70's (ver [12]), incluye a las funciones polinomiales, a las exponenciales, a las logarítmicas y a las trigonométricas en intervalos acotados, entre otras.

Un resultado fundamental probado por Khovanskii (ver [11]) establece que un sistema de n ecuaciones dadas por funciones Pfaffianas en n variables definidas en un dominio $\mathcal{U} \subset \mathbb{R}^n$ tiene una cantidad finita de soluciones no degeneradas en $\mathcal{U}$ y que esta cantidad puede acotarse explícitamente en términos de parámetros sintácticos asociados al sistema. Este resultado, que puede verse como una generalización del teorema clásico de Bézout en geometría algebraica, permite probar diversas propiedades geométricas y topológicas de los conjuntos definidos a partir de funciones Pfaffianas (ver, por ejemplo, [8]). Desde el punto de vista algorítmico, en [7] puede encontrarse una síntesis de diversos resultados cuantitativos y de complejidad conocidos en el tratamiento de ecuaciones Pfaffianas, los cuales también se basan esencialmente en la cota de Khovanskii. Sin embargo, se cree que esta cota está lejos de ser óptima.

Esta tesis se enmarca en el análisis, desde el punto de vista efectivo, de sistemas de ecuaciones e inecuaciones dadas por cierta clase de funciones Pfaffianas en una variable y el estudio de la complejidad algorítmica de la resolución simbólica de problemas relacionados con estos sistemas. En este marco, se diseñan por un lado nuevos algoritmos eficientes y, por otra parte, se desarrollan herramientas teóricas que brindan información sobre los sistemas considerados que, a su vez, pueden dar lugar a la construcción de nuevas y mejores herramientas algorítmicas.

La clase de funciones considerada a lo largo de esta tesis es la de las funciones Pfaffianas de orden 1 en una variable, es decir, funciones de la forma $f(x) = F(x, \varphi(x))$ donde F es un polinomio en $\mathbb{Z}[X, Y]$ y φ es una función definida en un abierto $\mathcal{U} \subset \mathbb{R}$ que es solución de una ecuación diferencial del tipo $\varphi'(x) = \Phi(x, \varphi(x))$ con $\Phi \in \mathbb{Z}[X, Y]$ de grado positivo en Y.

En un primer paso, diseñamos un procedimiento simbólico para contar exactamente la cantidad de ceros que tiene una función Pfaffiana f de la clase considerada en un intervalo real cerrado y acotado I contenido en su dominio. El procedimiento que proponemos está basado en la construcción de una familia de secuencias de Sturm asociadas a la función f, la cual se obtiene por medio del cálculo del polinomios subresultantes (ver, por ejemplo [3]). La generalización de la secuencia de Sturm para funciones continuas en intervalos fue introducida en [9]. Métodos basados en secuencias de Sturm se aplicaron, por ejemplo, en [31] para el conteo de la cantidad de ceros reales de funciones de la forma $p(x) + q(x)e^{r(x)}$,

con $p, q, r \in \mathbb{Z}[X]$ y luego, en [15], para las funciones del tipo $F(x, e^x)$, donde $F \in \mathbb{Z}[X, Y]$, casos particulares de las funciones consideradas en esta tesis. Como es usual en la literatura, asumimos la existencia de un *oráculo* (una caja negra que siempre da la respuesta correcta) para determinar el signo que una función Pfaffiana toma cuando se la evalúa en un número real algebraico. Como consecuencia de nuestra construcción de secuencias de Sturm, obtenemos una cota superior para la cantidad de ceros de una función de la clase considerada en un intervalo contenido en su dominio.

A continuación analizamos el problema algorítmico de determinar la consistencia de sistemas de ecuaciones e inecuaciones dadas por familias finitas de funciones Pfaffianas de orden 1 en una variable. El problema de decisión asociado a funciones de este tipo tiene especial relevancia por su conexión con la existencia de eliminación de cuantificadores en la teoría de primer orden sobre $\mathbb{R}$ aumentada con términos exponenciales. Este problema fue planteado por Tarski en 1948. En [24] esta pregunta fue respondida negativamente y, luego en [13], la decidibilidad de la teoría fue probada, suponiendo que la conjetura de Shanuel es cierta, usando un enfoque teórico no adecuado para implementar. Posteriormente, el problema de decisión y algunas cuestiones relacionadas fueron considerados desde el punto de vista algorítmico para fragmentos de la teoría de primer orden de los reales extendida con una función Pfaffiana particular (ver, por ejemplo, [19], [27], [15], [2], [29] y [1]). Más recientemente, en [16], se muestra un procedimiento de decisión para ciertas clases de fórmulas de primer orden que involucran polinomios con coeficientes enteros y ciertas funciones trascendentes específicas y, en [32], se propone un método de eliminación de cuantificadores para fórmulas que involucran una función exponencial. Ambos métodos están basados en aislar ceros reales de funciones trascendentes de una variable por medio de un algoritmo de bisección, pero sus complejidades teóricas no han sido analizadas; más aún, obtener estimaciones de la complejidad parece ser una tarea difícil.

En esta tesis, introducimos una noción de *secuencias de Sturm generalizadas* para funciones Pfaffianas de orden 1 y la aplicamos a la resolución efectiva del problema de la consistencia, extendiendo los algoritmos clásicos que lo resuelven para polinomios reales (ver, por ejemplo, [3]). Más generalmente, utilizamos estas técnicas en el desarrollo de un algoritmo simbólico que, dada una familia $f_1(x) = F_1(x, \varphi(x)), \ldots, f_s(x) = F_s(x, \varphi(x))$, donde φ es una función Pfaffiana de orden 1 y $F_1, \ldots, F_s$ son polinomios con coeficientes enteros, determina la lista de todas las condiciones de signo factibles sobre $f_1, \ldots, f_s$, es decir, la lista de $\sigma = (\sigma_1, \ldots, \sigma_s)$ con $\sigma_i \in \{<, =, >\}$ para $i = 1, \ldots, s$, tales que el sistema $f_1(x)\sigma_1 0, \ldots, f_s(x)\sigma_s 0$ define un conjunto no vacío. Este procedimiento sirve como punto de partida para el diseño de un algoritmo simbólico que resuelve el problema de decisión para fórmulas definidas por funciones de la clase considerada. Al igual que para el conteo de ceros, estos algoritmos presuponen la existencia de un oráculo que determina el signo de funciones Pfaffianas en números reales algebraicos.

Finalmente, trabajamos con una familia particular de funciones Pfaffianas para la que la determinación de signos puede llevarse a cabo algorítmicamente de manera efectiva. Más precisamente, estudiamos las funciones de la forma $f(x) = F(x, e^{h(x)})$, donde F y h son polinomios con coeficientes enteros, llamadas E-polinomios (ver [27] y [20]). El resultado central obtenido en este contexto es la construcción de una subrutina, con estimaciones explícitas de su complejidad, que determina el signo de un E-polinomio en un número real algebraico

representado por medio de su codificación de Thom. Esta subrutina se aplica para obtener algoritmos de complejidad doblemente exponencial que no requieren llamadas a oráculos para el conteo de la cantidad de ceros de E-polinomios en intervalos reales y para la resolución del problema de decisión de fórmulas que involucran estas funciones.

En el caso de E-polinomios, obtenemos además una cota superior explícita para el valor absoluto de sus ceros reales en términos de los grados y las alturas de los polinomios involucrados. Esto da una respuesta al *problema de la última raíz*, planteado en [25] para funciones de este tipo. Previamente, en [30], la existencia de tal cota fue establecida para términos exponenciales generales, pero aún cuando esta cota está dada por un argumento inductivo con un número computable de iteraciones, la cota no es explícita. Para $F \in \mathbb{Z}[X, Y]$, en [15] se exhibe un algoritmo que calcula cotas superiores para los ceros reales de funciones de la forma $F(x, e^x)$ y , más generalmente, en [16], para funciones de la forma $F(x, \mathrm{trans}(x))$ con $\mathrm{trans}(x) = e^x,\ \ln(x)$ o $\arctan(x)$.

Por último, introducimos la noción de codificación de Thom para un cero real de un E-polinomio de una variable que generaliza la conocida para números reales algebraicos y permite distinguir y comparar los ceros de un E-polinomio. Damos además un algoritmo que determina estas codificaciones y estimamos su complejidad.

Esta tesis está organizada de la siguiente manera:

En el Capítulo 1, enunciamos algunos resultados teóricos y algorítmicos relacionados con polinomios en una y varias variables que utilizaremos a lo largo del trabajo. En el Capítulo 2, presentamos la clase de funciones con las que trabajamos en la tesis y establecemos algunas de sus propiedades (Sección 2.1). La Sección 2.2 está dedicada al algoritmo que calcula la cantidad de ceros de estas funciones: introducimos primero la noción de secuencia de Sturm, mostramos un algoritmo para construirlas y aplicamos esta construcción para el diseño del algoritmo central de la sección. En la Sección 2.3 se definen y construyen las secuencias de Sturm generalizadas que nos permiten resolver el problema de decisión en la Sección 2.4. Finalmente, el Capítulo 3 contiene nuestros resultados para E-polinomios. Luego de enunciar algunas propiedades básicas (Sección 3.1), presentamos la subrutina que determina el signo de un E-polinomio en un número real algebraico (Sección 3.2). En las Secciones 3.3 y 3.4 aplicamos esta subrutina para construir algoritmos efectivos para el conteo de ceros de un E-polinomio y para la resolución del problema de decisión. La última sección de este capítulo (Sección 3.5) contiene los resultados sobre cotas y codificación de los ceros de un E-polinomio.

Capítulo 1

Preliminares

En esta tesis, trabajaremos con polinomios en una o dos variables con coeficientes en $\mathbb{Q}$ o en $\mathbb{Z}$, aunque eventualmente aparecerán situaciones que involucren polinomios con coeficientes en $\mathbb{R}$ y en $\mathbb{C}$. En la Sección 1.1 comenzaremos introduciendo algunos resultados y notaciones básicas sobre polinomios. Definiremos la secuencia de Sturm de dos polinomios de una variable y veremos cómo ésta se aplica al cálculo de la cantidad de raíces reales en un intervalo (acotado o no) de uno de los polinomios con alguna condición de signo sobre el otro. Definiremos y probaremos resultados que involucran ciertos parámetros relacionados con polinomios que servirán para el cálculo de complejidades de los algoritmos que utilizaremos a lo largo de este trabajo. En la Sección 1.2, enunciaremos algunos resultados algorítmicos básicos con sus correspondientes complejidades. Las Secciones 1.3 y 1.4 están destinadas a desarrollar conceptos que utilizaremos como herramientas algorítmicas a lo largo de este trabajo.

1.1. Polinomios: nociones y notaciones básicas.

Utilizaremos la notación $F', F'', \ldots, F^{(k)}$ para referirnos a las primeras k derivadas de un polinomio $F \in \mathbb{R}[X]$, $\deg(F)$ para referirnos a su grado y $\mathrm{cp}(F)$ para referirnos a su coeficiente principal. Dado un polinomio $F \in \mathbb{R}[X, Y]$ escribiremos $\frac{\partial F}{\partial X}$ y $\frac{\partial F}{\partial Y}$ para referirnos a las derivadas parciales y notaremos $\deg_X(F)$ y $\deg_Y(F)$ a su grado respecto de la variable X e Y respectivamente, mientras que la notación $\deg(F)$ se referirá al grado total de F.

Si A es un dominio de factorización única (en nuestro trabajo, generalmente será $A = \mathbb{Z}$ o $A = \mathbb{Z}[X]$), dados finitos elementos $a_1, \ldots, a_n \in A$, no todos nulos, notaremos $\gcd(a_1, \ldots, a_n)$ al divisor común mayor entre ellos. Si $F(X) = \sum_{i=0}^{n} a_i X^i \in A[X]$, el *contenido de* F es el elemento

$$\mathrm{cont}(F) := \gcd\big(a_0, a_1, \ldots, a_n\big) \in A.$$

Luego, todo $F \in A[X]$ se escribe de forma única como

$$F(X) = \mathrm{cont}(F).\bar{F}(X), \tag{1.1}$$

con $\mathrm{cont}(\bar{F}) = 1$. De esta forma, factorizar P en $A[X]$ es factorizar $\mathrm{cont}(F)$ en A y factorizar $\bar{F}$ en $A[X]$.

Teniendo en cuenta la notación de (1.1), se tiene que

$$\gcd(F,G) = \gcd\big(\mathrm{cont}(F), \mathrm{cont}(G)\big)\gcd(\bar{F},\bar{G}).$$

Finalmente, si A es un dominio íntegro y K su cuerpo de fracciones, dados $F, G \in A[X]$ polinomios con $G \neq 0$, al resto de dividir a F por G en $K[X]$ lo notaremos $\mathrm{Resto}(F,G)$.

1.1.1. Secuencia de Sturm

En esta sección definiremos la secuencia de Sturm de dos polinomios de una variable a coeficientes en $\mathbb{R}$. Veremos cómo esta secuencia se relaciona con la cantidad exacta de raíces reales del primer polinomio en un intervalo abierto, que puede estar acotado o no, y cómo además, esta cantidad se relaciona con los conjuntos de positividad y negatividad del segundo polinomio. Este resultado es el que generalizaremos en el capítulo 2.

Como es usual en la bibliografía, comenzaremos introduciendo la definición de secuencia de Sturm de un polinomio en $\mathbb{R}[X]$:

Definición 1.1 *Sea $F \in \mathbb{R}[X]$, $\deg(F) > 0$. Llamaremos* secuencia de Sturm de F *a la secuencia de polinomios $(r_0, \ldots, r_k)$ definida recursivamente de la siguiente manera: $r_0 = F$, $r_1 = F'$, $r_i = r_{i-1}g_i - r_{i-2}$ donde $g_i \in \mathbb{R}[X]$ y $0 \leq \deg(r_i) < \deg(r_{i-1})$ para $i = 2, \ldots, k$. La recursión finaliza cuando r_k es un divisor de r_{k-1}. Notar que r_k, salvo un factor constante, es el* $\gcd(F, F')$.

Generalizando esta definición, obtenemos la siguiente:

Definición 1.2 *Sean F y G dos polinomios de una variable con coeficientes en $\mathbb{R}$. Llamaremos* secuencia de Sturm de F y G *a la secuencia de polinomios $\mathbf{r} = (r_0, \ldots, r_k)$ definida recursivamente de la siguiente manera: $r_0 = F$, $r_1 = G$, $r_i = r_{i-1}g_i - r_{i-2}$ donde $g_i \in \mathbb{R}[X]$ y $0 \leq \deg(r_i) < \deg(r_{i-1})$ para $i = 2, \ldots, k$. La recursión finaliza cuando r_k es un divisor de r_{k-1}. Notar que r_k, salvo un factor constante, es el* $\gcd(F, G)$.

Definición 1.3 *Dada una secuencia $(\xi_0, \ldots, \xi_k)$ de elementos de $\mathbb{R}$, definimos la* cantidad de cambios de signo de la secuencia *como*

$$\#\left\{0 \leq i \leq k-1 \ / \ \xi_i.\xi_l < 0 \text{ con } l > i \text{ y } \xi_j = 0 \text{ para todo } i < j < l\right\}.$$

Si $a \in \mathbb{R}$, notaremos $v(\mathbf{r}, a)$ a la cantidad de cambios de signos de la secuencia de polinomios $\mathbf{r} = (r_0, \ldots, r_k)$ evaluada en a, es decir, la cantidad de cambios de signo de la secuencia $(r_0(a), \ldots, r_k(a))$.

Extendiendo esta noción a $+\infty$ y $-\infty$, notaremos $v(\mathbf{r}, +\infty)$ a la cantidad de cambios de signo de la secuencia $(\mathrm{cp}(r_0), \ldots, \mathrm{cp}(r_k))$ y $v(\mathbf{r}, -\infty)$ a la cantidad de cambios de signo de la secuencia $((-1)^{\deg(r_0)}\mathrm{cp}(r_0), \ldots, (-1)^{\deg(r_k)}\mathrm{cp}(r_k))$.

Ahora estamos en condiciones de enunciar una de las aplicaciones más importantes de estas secuencias: relacionar el número de raíces reales de un polinomio en un intervalo con la cantidad de cambios de signo de su secuencia de Sturm evaluada en los extremos de tal intervalo.

Teorema 1.4 *(ver [3, Theorem 2.62]) Sean $a, b \in \mathbb{R} \cup \{-\infty, +\infty\}$, $a < b$, $F \in \mathbb{R}[X]$ un polinomio que no se anula ni en a ni en b y $\mathbf{r}$ la secuencia de Sturm de F. Entonces $v(\mathbf{r}, a) - v(\mathbf{r}, b)$ es la cantidad de raíces reales distintas de F en el intevalo (a, b).*

Este teorema resulta ser un caso particular del teorema de Tarski. Antes de enunciarlo, introducimos la siguiente definición (ver [3, Notation 2.68]) :

Definición 1.5 *Sean $F, G \in \mathbb{R}[X]$ y $a, b \in \mathbb{R} \cup \{-\infty, +\infty\}$, $a < b$. Definimos el* indicador de Tarski *del polinomio F para G en (a, b), y lo notaremos* $\mathrm{TaQ}(F, G; a, b)$*, como el número*

$$\#\{x \in (a, b) \ / \ F(x) = 0 \wedge G(x) > 0\} - \#\{x \in (a, b) \ / \ F(x) = 0 \wedge G(x) < 0\}.$$

El indicador de Tarski de un polinomio F para G se puede calcular utilizando la secuencia de Sturm de F y $F'G$:

Teorema 1.6 *([3, Theorem 2.73]) Sean $F, G \in \mathbb{R}[X]$, $a, b \in \mathbb{R} \cup \{-\infty, +\infty\}$, $a < b$, ninguno de ellos raíz de F y sea $\mathbf{r}$ la secuencia de Sturm de F y $F'G$. Entonces*

$$\mathrm{TaQ}(F, G; a, b) = v(\mathbf{r}, a) - v(\mathbf{r}, b).$$

1.1.2. Medidas de polinomios y de números algebraicos

Algunos parámetros que van a aparecer en las complejidades de los algoritmos que consideraremos a lo largo de la tesis serán, no sólo cotas para el grado de los polinomios involucrados, sino parámetros relacionados con sus coeficientes. A lo largo de esta sección, los definiremos y obtendremos algunos resultados que utilizaremos en el cálculo de complejidad de los distintos algoritmos que aparecerán conforme avance este trabajo.

En algunos resultados, dado un polinomio $F(X) \in \mathbb{R}[X]$, será útil referirnos a la norma 2 del vector cuyas coordenadas son los coeficientes de F. Para ello, introducimos la notación $\|F\| = \big(\sum_{i=0}^{d} a_i^2\big)^{\frac{1}{2}}$, donde $F(X) = \sum_{i=0}^{d} a_i X^i$.

A continuación, definimos la medida de Mahler de un polinomio de una variable con coeficientes reales, introducida por Mahler en 1962.

Definición 1.7 *Sea $F \in \mathbb{R}[X]$ un polinomio de grado d. Supongamos que $F(X) = \sum_{i=0}^{d} a_i X^i$ y que $\alpha_1, \ldots, \alpha_d \in \mathbb{C}$ son sus raíces repetidas tantas veces como su multiplicidad lo indique.*

Definimos la medida de Mahler de P *como el número*

$$M(F) = |a_d| \prod_{j=1}^{d} \max\{1, |\alpha_j|\}.$$

Definimos además la altura de F *como el número*

$$H(F) = \max\{|a_i| : i = 0, \ldots, d\},$$

y más en general, si $F \in \mathbb{R}[X,Y]$, $F(X,Y) = \sum_{i,j\geq 0; i+j\leq d} a_{ij}X^iY^j$, *definimos la altura de* F *como el número*

$$H(F) = \max\{|a_{ij}| : \ i+j \leq d\}.$$

La medida de Mahler verifica trivialmente la fórmula

$$M(FG) = M(F)M(G) \ \forall F, G \in \mathbb{R}[X]. \tag{1.2}$$

Estas nociones se definen también para números algebraicos:

Definición 1.8 *Sea* α *un número algebraico sobre* $\mathbb{Q}$ *y* $m_\alpha \in \mathbb{Z}[X]$ *su polinomio minimal (definido salvo signo). Llamaremos* grado de α *al número* $\deg(\alpha) = \deg(m_\alpha)$, medida de Mahler *de* α *al número* $M(\alpha) = M(m_\alpha)$ *y* altura de α *al número* $H(\alpha) = H(m_\alpha)$.

Si lo único que sabemos de un número algebraico α es que es raíz de un polinomio en $\mathbb{Z}[X]$ dado, igualmente podemos acotar su grado y su altura. Para esto, usaremos los siguientes resultados:

Lema 1.9 *Sea* $F \in \mathbb{R}[X]$. *Entonces:*

i) $H(F) \leq 2^{\deg(F)}M(F)$ *y*

ii) $M(F) \leq (\deg(F)+1)^{\frac{1}{2}}H(F)$.

Demostración. Supongamos que $F(X) = \sum_{i=0}^{d} a_iX^i$ con $a_d \neq 0$.

i) Si $\alpha_1, \ldots, \alpha_d \in \mathbb{C}$ son todas sus raíces, se tiene que la relación entre los coeficientes de F y sus raíces es

$$a_{d-k} = (-1)^k\Big(\sum_{1\leq i_1<\ldots<i_k\leq d} \alpha_{i_1}\ldots\alpha_{i_k}\Big)a_d \quad \forall k = 1, \ldots, d,$$

por lo que $|a_{d-k}| \leq \binom{d}{k}M(F) \leq 2^dM(F)$, $\forall k = 1, \ldots, d$. Luego, $H(F) \leq 2^dM(F)$.

ii) Por [3, Proposition 10.9] se tiene que $M(F) \leq \|F\|$. Considerando la desigualdad

$$\|F\| \leq (\deg(F)+1)^{\frac{1}{2}}H(F), \tag{1.3}$$

se obtiene la cota propuesta.

□

Lema 1.10 *Sea* α *un número algebraico sobre* $\mathbb{Q}$ *y* $F \in \mathbb{Z}[X]$ *tal que* $F(\alpha) = 0$, *entonces:*

i) $\deg(\alpha) \leq \deg(F)$ *y*

ii) $H(\alpha) \leq 2^{\deg(F)}(\deg(F)+1)^{\frac{1}{2}}H(F)$.

Demostración. Como $F \in \mathbb{Z}[X]$ y $F(\alpha) = 0$ se tiene que el polinomio minimal de α, m_α, divide a F. Luego $i)$ se verifica trivialmente. Para probar $ii)$, usamos el Lema 1.9 $i)$ aplicado al polinomio minimal de α y luego el hecho de que $M(\alpha) \leq M(F)$ (pues las raíces de F con módulo mayor que 1 incluyen a las de m_α con módulo mayor que 1), obteniendo que

$$H(\alpha) \leq 2^{\deg(\alpha)} M(\alpha) \leq 2^{\deg(F)} M(F).$$

Aplicando a continuación el Lema 1.9 $ii)$, se obtiene que

$$H(\alpha) \leq 2^{\deg(F)} (\deg(F) + 1)^{\frac{1}{2}} H(F)$$

como se quería probar. □

Algunas propiedades que verifica la altura de polinomios en una y dos variables y que necesitaremos para el cálculo de complejidades se resumen en la siguiente proposición:

Proposición 1.11 *Sean $G_1, \ldots, G_n \in \mathbb{Z}[X]$ y $F_1, \ldots, F_m \in \mathbb{Z}[X, Y]$. Entonces:*

a) $H\big(\sum_{i=1}^{m} F_i\big) \leq \sum_{i=1}^{m} H(F_i)$.

b) $H\big(\frac{\partial F_1}{\partial X}\big) \leq \deg_X(F_1) H(F_1)$ *y* $H\big(\frac{\partial F_1}{\partial Y}\big) \leq \deg_Y(F_1) H(F_1)$.

c) $H(G_1 G_2) \leq (\text{mín}\{\deg(G_1), \deg(G_2)\} + 1) H(G_1) H(G_2)$.

d) $H\big(\prod_{i=1}^{n} G_i\big) \leq \prod_{i=1}^{n} H(G_i)(\deg(G_i) + 1)$.

e) $H\big(\prod_{i=1}^{m} F_i\big) \leq \prod_{i=1}^{m} H(F_i)(\deg(F_i) + 1)^2$.

Demostración.

a) y *b)* son triviales.

c) Es consecuencia de que el coeficiente del polinomio $G_1 G_2$ de potencia X^i es sumar a lo sumo mín $\{\deg(G_1), \deg(G_2)\} + 1$ términos que son productos de un coeficiente de G_1 por uno de G_2.

d) Se obtiene por inducción en n. Si $n = 2$, por el ítem $c)$, vale que

$$H(G_1 G_2) \leq (\text{mín}\{\deg(G_1), \deg(G_2)\} + 1) H(G_1) H(G_2) \leq$$

$$\leq (\deg(G_1) + 1)(\deg(G_2) + 1) H(G_1) H(G_2).$$

Si suponemos que vale para n, obtenemos que

$$H\big(\prod_{i=1}^{n+1} G_i\big) = H\big(G_{n+1} \prod_{i=1}^{n} G_i\big) \leq$$

$$\leq \big(\text{mín}\{\deg(G_{n+1}), \sum_{i=1}^{n} \deg(G_i)\} + 1\big) H(G_{n+1}) H(\prod_{i=1}^{n} G_i) \leq$$

$$\leq \big(\text{mín}\{\deg(G_{n+1}), \sum_{i=1}^{n} \deg(G_i)\} + 1\big) H(G_{n+1}) \prod_{i=1}^{n} H(G_i)(\deg(G_i) + 1) \leq$$

$$\leq \prod_{i=1}^{n+1} H(G_i)(\deg(G_i) + 1).$$

e) Supongamos que $F_i(X,Y) = \sum_{j=0}^{d_{iY}} a_{ij}(X)Y^j$. Luego,

$$\prod_{i=1}^{m} F_i(X,Y) = \sum_{t=0}^{d_{1Y}+\ldots+d_{mY}} \Big(\sum_{t=j_1+\ldots+j_m} a_{1j_1}(X).\ldots.a_{mj_m}(X)\Big) Y^t. \tag{1.4}$$

Por el ítem *d)* , se tiene que

$$H(a_{1j_1}(X).\ldots.a_{mj_m}(X)) \leq \prod_{i=1}^{m} H(a_{ij_i})(\deg(a_{ij_i}) + 1) \leq \prod_{i=1}^{m} H(F_i)(\deg_X(F_i) + 1).$$

Por otro lado, la suma sobre la descomposición de t en (1.4), tiene a lo sumo

$$\prod_{i=1}^{m}(d_{iY} + 1) = \prod_{i=1}^{m}(\deg_Y(F_i) + 1)$$

sumandos. Luego,

$$H(\prod_{i=1}^{m} F_i) \leq \left[\prod_{i=1}^{m} H(F_i)(\deg_X(F_i) + 1)\right] \left[\prod_{i=1}^{m}(\deg_Y(F_i) + 1)\right].$$

Usando que $(\deg_X(F_i) + 1)(\deg_Y(F_i) + 1) \leq (\deg(F_i) + 1)^2$ para todo $i = 1, \ldots, m$, se obtiene la cota deseada.

□

Observación 1.12 *Con las hipótesis de la Proposición 1.11, notar que vale que*

$$H(G_1F_1) \leq \big(\text{mín}\{\deg(G_1), \deg_X(F_1)\} + 1\big) H(G_1)H(F_1), \tag{1.5}$$

puesto que si $F_1(X,Y) = \sum_{j=0}^{d} a_j(X)Y^j$*, se tiene que* $G_1(X)F_1(X,Y) = \sum_{j=0}^{d} G_1(X)a_j(X)Y^j$ *y por lo tanto,* $H(G_1F_1) \leq \max\{H(G_1a_j) : j = 0, \ldots, \deg_Y(F_1)\}$*. Aplicando el ítem c) se obtiene la cota deseada.*

Será útil también considerar una cota para la altura de un factor de un polinomio en función de la altura del polinomio. Para ello citamos el siguiente lema previo:

Lema 1.13 *([17, Proposition 2.1.13]) Sean $F, G \in \mathbb{Z}[X]$ no nulos, $G(X) = \sum_{i=0}^{\deg(G)} a_i X^i$ un divisor de F en $\mathbb{Z}[X]$. Entonces* $\sum_{j=0}^{\deg(G)} |a_i| \leq 2^{\deg(G)} \|F\|$.

Del lema anterior y de la desigualdad (1.3) se deduce que

$$H(G) \leq 2^{\deg(G)} \sqrt{\deg(F)+1}\; H(F).$$

De esta desigualdad obtenemos una cota para la altura del contenido de un polinomio en $\mathbb{Z}[X][Y]$.

Corolario 1.14 *Sea $F \in \mathbb{Z}[X,Y]$ de grado total d y $\mathrm{cont}(F)$ el contenido de F como polinomio de variable Y. Entonces $H(\mathrm{cont}(F)) \leq 2^d \sqrt{d+1}\; H(F)$.*

En [14, Section 5], se prueba que si $F = \prod_{i=1}^{s} F_i$, $F_i \in \mathbb{R}[X,Y]$ para todo $i = 1, \dots s$, $d_X = \sum_{i=1}^{s} \deg_X(F_i)$ y $d_Y = \sum_{i=1}^{s} \deg_Y(F_i)$, se tiene que

$$\prod_{i=1}^{s} H(F_i) \leq 2^{d_X+d_Y} \big((d_X+1)(d_Y+1)\big)^{\frac{1}{2}} H(F).$$

De aquí se deduce trivialmente una cota superior de la altura de un factor de un polinomio en función de la altura del polinomio en el caso de polinomios en $\mathbb{Z}[X,Y]$:

Proposición 1.15 *Sean $G, Q \in \mathbb{Z}[X,Y]$ no nulos, $F = GQ$, $\deg(F) \leq d$. Entonces*

$$H(G) \leq 4^d (d+1) H(F).$$

1.1.3. Tamaño y separación de raíces

Para desarrollar algoritmos efectivos necesitaremos, dado un polinomio en $\mathbb{Z}[X]$, conocer un intervalo acotado, con extremos racionales, que contenga todas sus raíces reales, así como también acotar inferiormente la distancia de separación entre ellas, sin necesidad de conocerlas. Es por eso que introducimos los siguientes resultados. El primero, nos dará una cota superior para el tamaño de las raíces de un polinomio y, el segundo, una cota inferior para la distancia entre dos raíces distintas.

Lema 1.16 (ver [17, Corollary 2.5.22]) *Sea $F(X) = \sum_{i=0}^{d} a_i X^i \in \mathbb{R}[X]$, $a_d \neq 0$ y $\alpha \in \mathbb{C}$ una de sus raíces. Entonces*

$$|\alpha| < 1 + \max\left\{ \left| \frac{a_i}{a_d} \right| : 0 \leq i \leq d-1 \right\}.$$

En particular, notar que si $F \in \mathbb{Z}[X]$, para todo $\alpha \in \mathbb{C}$ raíz de F se tiene que

$$|\alpha| < 1 + H(F).$$

Teorema 1.17 (ver [17, Theorem 2.7.2]) *Sea $F \in \mathbb{Z}[X]$ un polinomio de grado $d \geq 2$ y $\alpha_1, \ldots, \alpha_d$ todas sus raíces complejas. Entonces*

$$\text{mín}\{\ |\alpha_i - \alpha_j|\ :\ \alpha_i \neq \alpha_j\ \} > d^{-\frac{d+2}{2}}\, \|F\|^{1-d}.$$

Además, por (1.3), se tiene que

$$\text{mín}\{\ |\alpha_i - \alpha_j|\ :\ \alpha_i \neq \alpha_j\} > d^{-\frac{d+2}{2}} \left(\sqrt{(d+1)}H(F)\right)^{1-d}. \tag{1.6}$$

1.2. Algoritmos y complejidades

Un algoritmo es un procedimiento que toma un conjunto finito de datos, que llamaremos *input*, y luego de una cantidad finita de pasos, en cada uno de los cuales se efectúan ciertas operaciones prefijadas, produce otro conjunto finito de datos, que llamaremos *output*.

En este trabajo, las operaciones prefijadas que consideraremos serán cada suma, producto y comparación que se realiza en $\mathbb{Q}$.

Uno de los problemas a considerar es cómo ingresar los datos en la máquina. Se necesita para ello una codificación de los datos del input. Por ejemplo, para codificar un polinomio en una variable de grado d con coeficientes en $\mathbb{Q}$ podremos pensar en la $(d+1)-$upla de sus coeficientes, es decir, el $i-$ésimo elemento de la upla corresponde al coeficiente en el monomio de grado $i-1$; o de manera análoga, podremos representar un polinomio de grado d en n variables teniendo en cuenta que hay $N = \binom{d+n}{n}$ monomios de grado a lo sumo d en n variables, estableciendo un orden entre estos monomios: el polinomio puede codificarse como la $N-$upla de sus coeficientes (incluyendo los nulos) en este orden. Si en cambio buscamos una codificación para un número real algebraico, por ejemplo el número $\sqrt{2}$, podríamos pensar en un polinomio con coeficientes en $\mathbb{Q}$ que lo tenga como raíz, digamos $Q(X) = X^2 - 2$, pero todavía faltaría diferenciar sus dos raíces. Para esto podríamos pensar en el signo de la primera derivada, que distingue a la raíz positiva de la negativa. Esta idea para codificar números reales algebraicos es la que utilizaremos en esta tesis, junto con la posibilidad de decidir, a partir de esta codificación, si se verifican o no ciertas condiciones de signos (ver Definición 1.24 y Proposición 1.25).

La *complejidad* de un algoritmo es una noción que se define para evaluar la eficiencia con la que se lleva a cabo la ejecución de éste en función de los datos ingresados como input. En esta tesis, al hablar de la complejidad de un algoritmo estaremos acotando la cantidad máxima de operaciones y de comparaciones que se realizan en $\mathbb{Q}$. Así, realizar una operación o comparación en $\mathbb{Q}$ tendrá complejidad 1.

Las distintas clases de complejidad se definen como una cota asintótica de estas cantidades. Más formalmente, definimos en el conjunto de las funciones de una variable real los conjuntos

$$O\big(g(x)\big) := \big\{ f(x) \ / \text{ existen } x_0, c > 0 \text{ tales que } \forall x \geq x_0 \ \ |f(x)| \leq c\,|g(x)| \,\big\}.$$

Diremos que un algoritmo tiene complejidad de orden $O(g(x))$ si la cantidad de operaciones y comparaciones en $\mathbb{Q}$ es un elemento $f(x)$ del conjunto $O(g(x))$, donde x representa una medida de los datos del input. En este caso, se permitirá la notación, $f(x) = O(g(x))$. Además, si para dos funciones $g_1(x)$ y $g_2(x)$ se tiene que $O(g_1(x)) \subset O(g_2(x))$, usaremos la notacíón $O(g_1(x)) \leq O(g_2(x))$.

Como es usual, para el cálculo de complejidades utilizaremos el logaritmo en base 2 que notaremos $\log(x)$. Además, siempre que consideremos una cota superior para el grado de un polinomio dado, supondremos que es un número natural.

Una cantidad que aparecerá en las complejidades de muchos algoritmos es

$$M(d) := d \log(d) \log\log(d).$$

Introducimos a continuación algunos resultados algorítmicos, junto con sus respectivas complejidades, que vamos a utilizar a lo largo de esta tesis:

- Derivar y sumar polinomios en $\mathbb{Q}[X]$ de grados acotados por d puede calcularse con un algoritmo que requiere $O(d)$ operaciones (ver [3, Algorithm 8.1]), mientras que el producto puede calcularse con un algoritmo que requiere $O(M(d))$ (ver [26, Theorem 8.23]).

- Evaluar un polinomio con coeficientes en $\mathbb{Q}$, de grado acotado por d, en un elemento $b \in \mathbb{Q}$ puede realizarse con un algoritmo de complejidad $O(d)$ (ver [3, Algorithm 8.14]).

- El divisor común mayor entre polinomios en $\mathbb{Q}[X]$ de grados acotados por d puede calcularse, vía el Algoritmo rápido de Euclides, con $O(d \log^2(d))$ operaciones en $\mathbb{Q}$ (ver [26, Theorem 11.5 - Definition 8.26]).

- Dados $d+1$ puntos en $\mathbb{Q}$, encontrar su polinomio interpolador de grado menor o igual que d puede realizarse con un algoritmo que requiere $O(M(d)\log(d))$ operaciones (ver [26, Corollary 10.12]).

- El determinante de una matriz en $\mathbb{Q}^{d\times d}$ puede calcularse con un algoritmo de complejidad $O(d^\omega)$, donde $\omega < 2{,}376$ (ver, por ejemplo, [26, Chapter 12]).

Una aplicación del Algoritmo rápido de Euclides antes mencionado es calcular la secuencia de Sturm de dos polinomios con coeficientes en $\mathbb{Q}$:

Proposición 1.18 *Sean $F, G \in \mathbb{Q}[X]$ polinomios de grados acotados por d. La secuencia de Sturm de F y G puede ser calculada con una complejidad de orden $O(d \log^2(d))$.*

Teorema 1.19 *Sean $F, G \in \mathbb{Q}[X]$ polinomios de grados acotados por d y sea $I = (a, b)$ un intervalo con extremos en $\mathbb{Q} \cup \{-\infty, +\infty\}$. El indicador de Tarski de F para G en I, $\mathrm{TaQ}(F, G; a, b)$, puede ser calculado con una complejidad de orden $O(d \log^2(d))$.*

Demostración. Una vez calculada la Secuencia de Sturm de F y $F'G$, por ejemplo con el algoritmo de la Proposición 1.18, sólo hay que evaluarla en a y b, lo cual no modifica la complejidad. □

Dado un polinomio en $\mathbb{Z}[X]$, necesitaremos aproximar sus raíces por números racionales para lo cual hallaremos intervalos disjuntos de longitud prefijada con extremos racionales que las contengan. Para esto, daremos un algoritmo que utiliza secuencias de Sturm.

Proposición 1.20 *Sea $F \in \mathbb{Z}[X]$ de grado acotado por d y $\theta \in \mathbb{Q}$. Existe un algoritmo que calcula finitos intervalos disjuntos, con extremos racionales, de longitud menor o igual que θ y tal que todos contienen alguna raíz real de F y todas ellas están contenidas en alguno. La complejidad de este algoritmo es $O(d^3 \log(\frac{H(F)}{\theta}))$.*

Demostración. El algoritmo construye de forma recursiva finitos intervalos. Se comienza con el intervalo $J = (-(1 + H(F)), 1 + H(F)]$ que, por el Lema 1.16, contiene a todas las raíces reales de F. En cada paso intermedio, dado un intervalo $J = (a, b]$ con $\{x \,/\, F(x) = 0\} \cap J \neq \emptyset$ y $|J| > \theta$, se procede de la siguiente manera:

- Sea $c = \frac{a+b}{2}$ y $J_r = (c, b]$.
- Si $F(c) \neq 0$, tomar $J_l = (a, c]$.
- Si $F(c) = 0$ y $c - \theta > a$, sean $I = (c - \theta, c]$ y $J_l = (a, c - \theta]$. Si $F(c) = 0$ y $c - \theta \leq a$, tomar $I = (a, c]$. Observar que, en cualquier caso, I contiene una raíz real de F y tiene longitud de a lo sumo θ.
- Determinar, para cada uno de los intervalos J_r y J_l, si F tiene una raíz real o no en ese intervalo. Guardar los intervalos que contienen la raíz de F.

La recursión finaliza cuando todos los intervalos tienen longitud a lo sumo θ. El output consiste de todos los intervalos de longitud a lo sumo θ que contienen las raíces de F, incluyendo los intervalos I que aparecen en los pasos intermedios.

Para determinar si F tiene una raíz real en un intervalo dado, se usa la secuencia de Sturm de F y F' (ver Teorema 1.4), la cual puede calcularse con complejidad $O(d \log^2(d))$ (ver Proposición 1.18).

En cada paso de la recursión, guardamos a lo sumo d intervalos junto con las cantidad de cambios de signo de la secuencia de Sturm evaluada en cada uno de los extremos de los intervalos. Para cada uno de esos intervalos, el procedimiento de arriba requiere de a lo sumo $2d + 1$ evaluaciones adicionales de polinomios de grado a lo sumo d. Luego, la complejidad de cada paso recursivo es de orden $O(d^3)$.

Como la longitud de los intervalos del k-ésimo paso es de a lo sumo $\frac{1+H(F)}{2^{k-1}}$, la cantidad de pasos es a lo sumo $1 + \lceil \log(\frac{1+H(F)}{\theta}) \rceil$.

Se concluye que la complejidad total es de orden $O(d^3 \log(H(F)/\theta))$. □

1.3. Condiciones de signo y codificación de Thom

Consideremos la función signo $\mathrm{sg} : \mathbb{R} \to \{1, 0, -1\}$, definida por la fórmula

$$\mathrm{sg}(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x = 0 \\ -1 & \text{si } x < 0. \end{cases}$$

Definición 1.21 *Sea $\Sigma = \{F_1, \ldots, F_s\}$ un conjunto finito de polinomios en $\mathbb{R}[X]$. Una* condición de signo factible de Σ *sobre un conjunto $Z \subset \mathbb{R}$, es una s-upla $(\sigma_1, \ldots, \sigma_s) \in \{-1, 0, 1\}^s$ tal que*

$$\{x \in Z \ / \ \mathrm{sg}(F_i(x)) = \sigma_i \ \forall i = 1, \ldots, s\} \neq \emptyset.$$

Un tipo particular de conjunto Z con el que nos interesará trabajar es el conjunto de raíces de un polinomio, para el cual vale el siguiente resultado:

Teorema 1.22 *(ver [18, Corollary 2]) Dados $F_0, F_1, \ldots, F_s \in \mathbb{Q}[X]$, $F_0 \neq 0$, de grados acotados por d, todas las condiciones de signo factibles de $F_1, \ldots, F_s$ sobre $\{x \in \mathbb{R} \ / \ F_0(x) = 0\}$ pueden ser computadas con $O(sd^2 \log^3(d))$ operaciones. Más aún, si F_0 tiene m raíces en $\mathbb{R}$, estas condiciones de signo pueden ser computadas con complejidad de orden*

$$O(smd \log(m) \log^2(d)).$$

El algoritmo consiste en realizar un procedimiento recursivo que, en cada paso $i = 1, \ldots, s$, computará las condiciones de signo factibles para los polinomios $F_1, \ldots, F_i$ sobre

$$Z = \{x \in \mathbb{R} \ / \ F_0(x) = 0\}$$

por medio del cálculo de indicadores de Tarski de polinomios apropiados y de la resolución de un sistema de ecuaciones lineales. La demostración se basa en el procedimiento dado en [5] combinado con una forma más eficiente de resolver ciertos sistemas lineales, propuesta en [18]. La idea de la demostración es la siguiente.

Como $m := \#Z \leq \deg(F_0)$, hay a lo sumo m condiciones de signo factibles, puesto que todo $x \in Z$ verifica la condición de signo $(\mathrm{sg}(F_1(x)), \ldots, \mathrm{sg}(F_s(x)))$, que podrían ser iguales o no para dos elementos distintos de Z.

Para $i = 1$, hay tres condiciones de signo posibles:

$$\{x \in Z \mid F_1(x) > 0\}, \ \{x \in Z \mid F_1(x) = 0\} \ \ y \ \ \{x \in Z \mid F_1(x) < 0\}.$$

Llamemos c^+, c^0 y c^- a los cardinales de estos conjuntos respectivamente. Claramente, las condiciones de signo factibles de F_1 sobre Z serán las que correspondan a los conjuntos de cardinal mayor o igual que 1. Estos cardinales verifican el siguiente sistema lineal de ecuaciones:

$$\begin{cases} c^0 + c^+ + c^- = \mathrm{TaQ}(F_0, 1) \\ \quad\ c^+ - c^- = \mathrm{TaQ}(F_0, F_1) \\ \quad\ c^+ + c^- = \mathrm{TaQ}(F_0, F_1^2), \end{cases}$$

donde todos los indicadores de Tarski son en $\mathbb{R}$.

Este paso requiere de:

- Calcular 3 indicadores de Tarski de polinomios de grado a lo sumo $2d$.

- Resolver el sistema lineal de tamaño 3×3.

Suponiendo finalizado el paso $i-1$, se obtienen a lo sumo m condiciones de signo factibles de $F_1, \ldots, F_i$ sobre Z. Cada una de ellas dará lugar a tres posibles condiciones de signo para el paso i. En [5], se prueba que los cardinales de estos conjuntos forman la solución de un sistema lineal compatible determinado de tamaño a lo sumo $3m \times 3m$, en el cual aparecen involucrados ciertos indicadores de Tarski. Más aún, en [5, Lemma 2.3], se prueba que existen $F_{j_1}, \ldots, F_{j_t}$, $t \leq \log(m)$, tales que los indicadores de Tarski que aparecen en el sistema son del tipo $\mathrm{TaQ}(F_0, F_{j1}^{\alpha_1} \ldots F_{jt}^{\alpha_t})$ con $\alpha_l \in \{0, 1, 2\}$.

Por ello, en el paso i se requiere:

- Calcular a lo sumo $3m$ indicadores de Tarski (como los descriptos arriba) del tipo $\mathrm{TaQ}(F_0, P)$ con grado de P acotado por $2d\log(m)$. Estos indicadores pueden calcularse con complejidad de orden $O(d\log(m)\log^2(d))$, pues $m \leq d$ (ver Teorema 1.19).

- Resolver un sistema lineal de ecuaciones de tamaño a lo sumo $3m \times 3m$. Esto puede realizarse, por [18], con complejidad de orden $O(m^2)$.

Por lo tanto, el paso i puede hacerse con complejidad de orden: $O(md\log(m)\log^2(d))$.

Obtenemos entonces que la complejidad total es de orden $O(smd\log(m)\log^2(d))$.

Dado que vamos a necesitar codificar números reales algebraicos sobre $\mathbb{Q}$ para poder utilizarlos en los algoritmos y además, vamos a querer operar con ellos a partir de su codificación, introducimos el siguiente resultado:

Proposición 1.23 *(ver [3, Proposition 2.36]) Sea $F \in \mathbb{Q}[X]$ de grado d y $(\sigma_1, \ldots, \sigma_d) \in \{-1, 0, 1\}^d$. El conjunto $\left\{x \in \mathbb{R} \ / \ \mathrm{sg}(F^{(i)}(x)) = \sigma_i \ \forall i = 0, \ldots, d\right\}$ es vacío o un punto, o bien un intervalo abierto.*

De esta proposición se deduce el hecho de poder distinguir raíces reales de polinomios con coeficientes en $\mathbb{Q}$ según el signo de sus derivadas en dichas raíces. La Proposición 1.23 permite concluir que, si x es raíz de un polinomio $F \in \mathbb{Q}[X]$, se verifica

$$\left\{y \in \mathbb{R} \ / \ \mathrm{sg}(F^{(i)}(y)) = \mathrm{sg}(F^{(i)}(x)) \ \forall i = 0, \ldots, d\right\} = \{x\}\,.$$

Obtenemos entonces una forma de codificar números reales algebraicos a partir de un polinomio en $\mathbb{Q}[X]$ que lo tenga como raíz.

Definición 1.24 *Dado $F \in \mathbb{Q}[X]$ un polinomio de grado d y una raíz real x de F, llamaremos* codificación de Thom de *x como raíz de F, y notaremos $\sigma_F(x)$, a la $(d+1)-$upla*

$$\left(\mathrm{sg}(F(x)), \mathrm{sg}(F^{(1)}(x)), \ldots, \mathrm{sg}(F^{(d)}(x))\right) \in \{-1, 0, 1\}^{d+1}\,.$$

A partir de sus codificaciones de Thom, no sólo pueden diferenciarse raíces de un polinomio sino que pueden ordenarse de menor a mayor, como lo prueba el siguiente resultado:

Proposición 1.25 *Sea $F \in \mathbb{Z}[X]$ no nulo y x_1, x_2 dos raíces reales distintas de F.*

Sea $k = \max\{i \ / \ F^{(i)}(x_1) \neq F^{(i)}(x_2)\}$. Entonces $F^{(k+1)}(x_1) = F^{(k+1)}(x_2)$ y son distintos de cero. Además:

- *Si* $\operatorname{sg}(F^{(k+1)}(x_1)) = \operatorname{sg}(F^{(k+1)}(x_2)) = 1$*:* $F^{(k)}(x_1) < F^{(k)}(x_2) \iff x_1 < x_2$.
- *Si* $\operatorname{sg}(F^{(k+1)}(x_1)) = \operatorname{sg}(F^{(k+1)}(x_2)) = -1$*:* $F^{(k)}(x_1) > F^{(k)}(x_2) \iff x_1 < x_2$.

Demostración. Por definición de k, se tiene que $F^{(k+1)}(x_1) = F^{(k+1)}(x_2)$. Si fuesen iguales a 0 entonces $\sigma_{F^{(k+1)}}(x_1) \neq \sigma_{F^{(k+1)}}(x_2)$ (pues son las codificaciones de Thom de x_1 y x_2 respecto al polinomio $F^{(k+1)}$), pero las últimas $\deg(F) - k$ coordenadas de $\sigma_F(x_i)$ coinciden con $\sigma_{F^{(k+1)}}(x_i)$, para $i = 1, 2$, lo cual es un absurdo por definición de k. Además,

$$\left\{ x/ \ \operatorname{sg}(F^{(j)}(x_1)) = \operatorname{sg}(F^{(j)}(x)) \text{ para } j = k+1, \ldots, \deg(F) - 1 \right\}$$

contiene a x_1 y a x_2 y, por la Proposición 1.23, resulta ser un intervalo abierto no vacío. Como en este intervalo, $F^{(k+1)}$ no se anula ni cambia de signo, $F^{(k)}$ es estrictamente creciente o estrictamente decreciente, de lo que se sigue el resultado a probar. □

Dado un polinomio F en $\mathbb{Z}[X]$, las codificaciones de Thom de sus raíces reales pueden ser calculadas algorítmicamente. Más aún, pueden ordenarse para que correspondan a las raíces ordenadas de menor a mayor.

Teorema 1.26 *Dado $F \in \mathbb{Z}[X]$ un polinomio de grado $d \geq 1$, existe un algoritmo que calcula la lista ordenada $\sigma_F(x_1), \sigma_F(x_2), \ldots, \sigma_F(x_t)$, donde $x_1, \ldots, x_t$ son todas las raíces reales de F ordenadas de menor a mayor. La complejidad de este algoritmo es de orden $O(d^3 \log^3(d))$.*

Demostración. Se comienza aplicando el algoritmo del Teorema 1.22 para calcular las condiciones de signo factibles de los polinomios $F', F'', \ldots, F^{(d)}$ sobre $\{x \in \mathbb{R} \ / \ F(x) = 0\}$. Por la Proposición 1.23, hay exactamente tantas condiciones de signo factibles como raíces reales tenga F.

Supongamos que estas condiciones son $\sigma_F(r_1), \ldots, \sigma_F(r_t)$, donde $r_1, \ldots, r_t$ son las raíces reales de F. Para ordenar estas raíces de menor a mayor a partir de sus codificaciones de Thom, hacemos inducción en el conjunto $\{r_1, \ldots, r_k\}$ de la siguiente manera:

- Si $k = 2$, usando la Proposición 1.25, calcular $q_1 = \min\{r_1, r_2\}$ y $q_2 = \max\{r_1, r_2\}$.
- Supongamos ordenado de menor a mayor el conjunto $\{r_1, \ldots, r_k\}$, renombrando a sus elementos $q_1 < \ldots < q_k$. Hallar $j_0 = \min\{j \ / \ \min\{r_{k+1}, q_j\} = r_{k+1}\}$ utilizando la Proposición 1.25. Entonces $q_1 < \ldots < q_{j_0 - 1} < r_{k+1} < q_{j_0} < \ldots < q_k$.

Calcular cada mínimo requiere a lo sumo d comparaciones en $\mathbb{Q}$ y en el paso k se requiere calcular a lo sumo $k-1$ mínimos, lo que da a lo sumo $\displaystyle\sum_{k=2}^{d} (k-1)d = d\frac{(d-1)d}{2}$ comparaciones.

Luego, pueden ordenarse las raíces reales de F a partir de su codificación de Thom con complejidad de orden $O(d^3)$. Notar que la complejidad de ordenar de esta forma las codificaciones de Thom no cambia la complejidad de calcular dichas codificaciones.

La complejidad final de este algoritmo es de orden $O(d^3 \log^3(d))$. □

Proposición 1.27 *Sean $F, G \in \mathbb{Q}[X]$ polinomios de grados acotados por d y $\alpha \in \mathbb{R}$ una raíz de F. Entonces existe un algoritmo que, dado $\sigma_F(\alpha)$, calcula el signo de $G(\alpha)$ en tiempo $O(d^3 \log^3(d))$.*

Demostración. Dado que los signos de F y sus derivadas en α determinan a α, alcanza con aplicar el algoritmo del Teorema 1.22, tomando $s = d+1$, $F_0 = F$, $F_i = F^{(i)}$ para $i = 1, \ldots, d$ y $F_{d+1} = G$, y obtener así la única condición factible que comienza con $\sigma_F(\alpha)$. □

Notar que puede adaptarse el algoritmo del Teorema 1.26, sin cambiar la complejidad, para calcular la lista ordenada de las codificaciones de Thom sólo de las raíces de F incluidas en un intervalo dado $I = (a, b)$. Para elegir las raíces de F que están en I, basta calcular los signos de los polinomios $X - a$ y $X - b$ en las raíces de F.

1.4. Resultantes y subresultantes de polinomios

La resultante de dos polinomios de una variable es un concepto que aparece en varias ramas de la matemática, siendo clave en la Teoría de Eliminación.

Los polinomios subresultantes son una generalización de la resultante de dos polinomios y fueron introducidas de manera implícita por C.G. Jacobi (ver [10]) y posteriormente de forma más explícita por J.J. Sylvester bajo el nombre "prime derivative of the d-degree" (ver [22] y [21]). La terminología "subresultante" parece ser de fines de los 60. Estos polinomios permiten, entre otras cosas, determinar cuál es el grado exacto del divisor común mayor entre los polinomios originales antes de calcularlo, siendo una herramienta para obtenerlo de forma alternativa al uso del algoritmo de Euclides. Una ventaja de trabajar con estas subresultantes es la de obtener mejoras en la complejidad de calcular el divisor común mayor entre dos polinomios, ya que los tamaños de los coeficientes de los polinomios que aparecen son menores que los que aparecen en los restos sucesivos del algoritmo de Euclides.

En esta tesis, utilizaremos estos polinomios como una herramienta algorítmica para calcular secuencias de Sturm para funciones Pfaffianas de una clase particular. Es por ello que comenzaremos dando su definición para luego analizar algunas propiedades que verifican. Mostraremos además, un algoritmo concreto para calcularlos.

Definición 1.28 *Sean $F(Y) = \sum_{i=0}^{m} a_i Y^i$ y $G(Y) = \sum_{i=0}^{n} b_i Y^i$ polinomios en $A[Y]$, donde A es un dominio íntegro. Definimos la* resultante de F y G, *y notamos* $\mathrm{Res}(F, G)$, *al determinante de la matriz cuyas filas son las coordenadas de los polinomios $Y^{n-1}F, \ldots, F, Y^{m-1}G, \ldots, G$*

en la base $\{Y^{n+m-1}, \ldots, Y, 1\}$*, es decir, es el determinante de la matriz*

$$\begin{pmatrix}
a_m & a_{m-1} & \cdots & \cdots & \cdots & a_0 & 0 & \cdots & 0 \\
0 & \ddots & & & & & \ddots & \ddots & \vdots \\
\vdots & \ddots & & \ddots & & & & \ddots & 0 \\
0 & \cdots & 0 & a_m & \cdots & \cdots & \cdots & \cdots & a_0 \\
b_n & b_{n-1} & \cdots & \cdots & b_0 & \cdots & \cdots & \cdots & 0 \\
0 & \ddots & & & & \ddots & \ddots & & \vdots \\
\vdots & \ddots & \ddots & & & & \ddots & \ddots & \vdots \\
\vdots & & \ddots & \ddots & & & & \ddots & 0 \\
0 & \cdots & \cdots & 0 & b_n & \cdots & \cdots & \cdots & b_0
\end{pmatrix}.$$

La motivación para definir y trabajar con resultantes radica en la validez del siguiente resultado, del cual puede hallarse una demostración en [3, Proposition 4.16]:

Proposición 1.29 *Sea A un dominio íntegro, K su cuerpo de fracciones y $F, G \in A[Y]$. Entonces:*

$$\operatorname{Res}(F, G) = 0 \iff F \text{ y } G \text{ tienen un factor común en } K[Y].$$

Para el caso particular en que el dominio íntegro sea $\mathbb{Z}[X]$ se obtiene el siguiente resultado:

Proposición 1.30 *Sean $F, G \in \mathbb{Z}[X, Y]$ y $R(X) = \operatorname{Res}_Y(F(X, Y), G(X, Y)) \in \mathbb{Z}[X]$ el polinomio resultante entre F y G respecto de la variable Y. Entonces:*

1) $\deg(R) \leq \deg_X(F)\deg_Y(G) + \deg_X(G)\deg_Y(F)$.

2) *Si $D \geq \max\{1, \deg_X(F)\deg_Y(G) + \deg_X(G)\deg_Y(F)\}$, R puede ser calculado algorítmicamente con complejidad de orden*

$$O\Big(D(\deg_Y(F) + \deg_Y(G))^\omega + M(D)\log(D)\Big).$$

Demostración.

1) El polinomio R resulta ser el determinante de una matriz de tamaño $(\deg_Y(F) + \deg_Y(G)) \times (\deg_Y(F) + \deg_Y(G))$ cuyos coeficientes son polinomios en $\mathbb{Z}[X]$ de grado menor o igual que $\deg_X(F)$ en las primeras $\deg_Y(G)$ filas y de grado menor o igual que $\deg_X(G)$ en las últimas $\deg_Y(F)$ filas, lo cual implica que $\deg(R) \leq \deg_Y(F)\deg_X(G) + \deg_Y(G)\deg_X(F)$.

2) Para calcular el polinomio R, se puede interpolar (evaluando en la variable X) en $D+1$ puntos. Esto requiere hacer $O(D+1)$ evaluaciones en F y en G en la variable X, calcular $D+1$ determinantes de matrices de tamaño $\deg_Y(F) + \deg_Y(G)$ con coeficientes en $\mathbb{Q}$ y usar el algoritmo de interpolación mencionado en la Sección 1.2. En total, se requiere del orden de $O\Big((D+1)(\deg_Y(F) + \deg_Y(G))^\omega + D\log^2(D)\log(\log(D))\Big)$ operaciones, lo que nos da una complejidad de $O\Big(D(\deg_Y(F) + \deg_Y(G))^\omega + M(D)\log(D)\Big)$.

□

Las siguientes definiciones se introducen para cualquier dominio íntegro A, aunque en nuestro caso, trabajaremos siempre sobre $\mathbb{Z}$ o sobre $\mathbb{Z}[X]$.

Definición 1.31 *Sean $F, G \in A[Y]$ de grados m y n respectivamente. Supongamos $n < m$. Para cada $0 \leq i \leq n$ definimos el $i-$*ésimo coeficiente subresultante de F y G, *y lo notaremos* $\mathrm{sRes}_i(F,G)$, *como el determinante de la matriz cuadrada obtenida de las primeras $n+m-2i$ columnas de la matriz de tamaño $(n+m-2i)\times(n+m-i)$ cuyas filas son las coordenadas de los polinomios $Y^{n-i-1}F, \ldots, F, G, \ldots, Y^{m-i-1}G$ en la base $\left\{Y^{n+m-i-1}, \ldots, Y, 1\right\}$, es decir que, si $F(Y) = \sum_{j=0}^{m} a_j Y^j$ y $G(Y) = \sum_{j=0}^{n} b_j Y^j$, $\mathrm{sRes}_i(F,G)$ es el determinante de la matriz que se obtiene al tomar las primeras $n+m-2i$ columnas de la matriz:*

$$A_i = \begin{pmatrix} a_m & a_{m-1} & \cdots & \cdots & \cdots & a_0 & 0 & 0 \\ 0 & \ddots & & & & & \ddots & 0 \\ \vdots & \ddots & a_m & \cdots & \cdots & \cdots & \cdots & a_0 \\ \vdots & & 0 & b_n & \cdots & \cdots & \cdots & b_0 \\ \vdots & \iddots & \iddots & & & & \iddots & \vdots \\ 0 & b_n & \ldots & \ldots & \ldots & b_0 & \ldots & 0 \\ b_n & \ldots & \ldots & \ldots & b_0 & 0 & \ldots & 0 \end{pmatrix}. \tag{1.7}$$

Por convención, se extiende esta definición para $i = -1$ y $n+1 \leq i \leq m$ de la siguiente manera:

$$\mathrm{sRes}_{-1}(F,G) = 0$$
$$\mathrm{sRes}_m(F,G) = cp(F),$$
$$\mathrm{sRes}_{m-1}(F,G) = cp(G),$$
$$\mathrm{sRes}_i(F,G) = 0 \text{ si } n < i < m-1.$$

Observación 1.32 *Notar que si $i = n$ en la matriz no hay filas con coeficientes de F, y por lo tanto,*

$$\mathrm{sRes}_n(F,G) = (-1)^{\frac{(m-n)(m-n-1)}{2}} \mathrm{cp}(G)^{m-n}.$$

Por otro lado, si $i = 0$ se tiene que

$$\mathrm{sRes}_0(F,G) = (-1)^{\frac{m(m-1)}{2}} \mathrm{Res}(F,G).$$

Notar que, dados $F, G \in A[Y]$ de grados m y n respectivamente, $m > n$, $\mathrm{sRes}_i(F,G)$ es un elemento del anillo A para todo $0 \leq i \leq m$. A continuación, vamos a definir los polinomios subresultantes entre F y G, que son polinomios en $A[Y]$.

Definición 1.33 *Sean $F, G \in A[Y]$ de grados m y n respectivamente. Supongamos $n < m$. Para cada $0 \leq i \leq n$ definimos la $i-$*ésima subresultante de F y G, *y la notaremos* $\mathrm{SRes}_i(F,G)$, *como el determinante de la matriz de tamaño $(n+m-2i)\times(n+m-2i)$ cuyas filas*

son las primeras $n+m-2i-1$ *coordenadas de los polinomios* $Y^{n-i-1}F,\dots,F,G,\dots,Y^{m-i-1}G$ *en la base* $\{Y^{n+m-i-1},\dots,Y,1\}$ *y los elementos de la última columna son los polinomios* $Y^{n-i-1}F,\dots,F,G,\dots,Y^{m-i-1}G$*, es decir,* $\mathrm{SRes}_i(F,G)$ *es el determinante de la matriz que se obtiene al tomar las primeras* $n+m-2i-1$ *columnas de la matriz* A_i *en (1.7) y agregar como última columna*

$$\begin{pmatrix} Y^{n-i-1}F \\ \dots \\ F \\ G \\ \dots \\ Y^{m-i-1}G \end{pmatrix}.$$

Por convención, se extiende esta definición para $i=-1$ *y* $n+1\le i\le m$ *de la siguiente manera:*

$$\mathrm{SRes}_{-1}(F,G)=0$$
$$\mathrm{SRes}_m(F,G)=F,$$
$$\mathrm{SRes}_{m-1}(F,G)=G,$$
$$\mathrm{SRes}_i(F,G)=0 \text{ si } n<i<m-1.$$

Proposición 1.34 *Con la notación de las Definiciones 1.31 y 1.33, se tiene que:*

a) Si $\mathrm{SRes}_i(F,G)\neq 0$*, para algún* $i\le n$*, su grado es menor o igual que* i*.*

b) $\mathrm{sRes}_i(F,G)$ *es el coeficiente de* Y^i *en* $\mathrm{SRes}_i(F,G)$*, para* $i\le m$*.*

Demostración. Observemos primero que la fila k de la matriz A_i es:

- si $k=1,\dots,n-i$:

$$\big(\underbrace{0\ 0\ \dots\ 0}_{k-1\text{ veces}}\ \ a_m\ \ a_{m-1}\ \ \dots\ \ a_0\ \ \underbrace{0\ 0\ \dots\ 0}_{n-i-k\text{ veces}}\big).$$

Observemos que el elemento de la columna j, con $k\le j\le m+k$, es a_{m+k-j}.
En particular, si $j=n+m-2i$, el elemento es

$$\begin{cases} 0 & \text{si } n-i\ge k+1 \\ a_{k-n-2i} & \text{si } k\le n+m-2i\le m+k. \end{cases}$$

- si $k=n-i+1,\dots,n+m-2i$:

$$\big(\underbrace{0\ 0\ \dots\ \dots\ 0}_{n+m-k-2i\text{ veces}}\ \ b_n\ \ b_{n-1}\ \ \dots\ b_0\ \ \underbrace{0\ \dots\ \dots\ 0}_{k-n+i-1\text{ veces}}\big).$$

Observemos que el elemento de la columna j, con $n+m-2i-k+1\le j\le 2n+m-2i-k+1$, es $b_{2n-k+1+m-2i-j}$. En particular, si $j=n+m-2i$, el elemento es

$$\begin{cases} b_{n-k+1} & \text{si } k\le n+1 \\ 0 & \text{si no.} \end{cases}$$

Si para $0 \leq i \leq n$ suponemos que $\mathrm{SRes}_i(F,G)$ es el determinante de una matriz de columnas $C_1, \ldots, C_{n+m-2i}$, usando la multilinealidad del determinante se puede reemplazar la última columna por $C_{n+m-2i} - \sum_{j=1}^{n+m-2i-1} Y^{n+m-i-j} C_j$ que el determinante no cambia. Veamos que polinomio tiene la nueva columna en la fila k.

- Si $1 \leq k \leq n-2i-1$:

$$Y^{n-i-1-k+1}F - \sum_{j=k}^{m+k} Y^{n+m-i-j} a_{m+k-j} = Y^{n-i-k}F - \sum_{l=0}^{m} a_l Y^{n+l-i-k} = Y^{n-i-k}(0) = 0.$$

- Si $n-2i \leq k \leq n-i$:

$$Y^{n-i-1-k+1}F - \sum_{j=k}^{n+m-2i-1} Y^{n+m-i-j} a_{m+k-j} = Y^{n-i-k}F - \sum_{l=k+2i-n+1}^{m} a_l Y^{n+l-i-k} =$$

$$= Y^{n-i-k}\big(F - \sum_{l=k+2i-n+1}^{m} a_l Y^l\big),$$

que tiene grado menor o igual que $n-i-k+k+2i-n = i$.

- Si $n-i+1 \leq k \leq n+1$:

$$Y^{k+i-n-1}G - \sum_{j=n+m-2i-k+1}^{n+m-2i-1} Y^{n+m-i-j} b_{2n+m-j-2i-k+1} =$$

$$= Y^{k+i-n-1}G - \sum_{l=n-k+2}^{n} b_l Y^{k+i-n+l-1} = Y^{k+i-n-1}\big(G - \sum_{l=n-k+2}^{n} b_l Y^l\big),$$

que tiene grado menor o igual que $k+i-n-1+n-k+1 = i$.

- Si $n+2 \leq k \leq n+m-2i$:

$$Y^{k+i-n-1}G - \sum_{j=n+m-2i-k+1}^{2n+m-2i-k+1} Y^{n+m-i-j} b_{2n+m-j-2i-k+1} =$$

$$= Y^{k+i-n-1}G - \sum_{l=0}^{n} b_l Y^{k+i-n+l-1} = Y^{k+i-n-1}\big(0\big) = 0.$$

Luego, obtenemos que $\mathrm{SRes}_i(F,G)$ es el determinante de una matriz de coeficientes en A salvo en la última columna que solo tiene polinomios en $A[Y]$ de grado a lo sumo i, por lo que se verifica la cota propuesta en a).
Por otro lado, el coeficiente de la potencia Y^i del polinomio en la fila k es, por lo anterior:

$$\begin{cases} a_{k-n+2i} & \text{si } \ k = n-2i, \ldots, n-i \\ b_{n-k+1} & \text{si } \ k = n-i+1, \ldots, n+1 \\ 0 & \text{si no.} \end{cases}$$

Luego, utilizando la multilinealidad del determinante, el determinante de esta matriz resulta ser $\mathrm{SRes}_i(F,G) = \det(M_i)Y^i + \det(N_i)$, donde M_i es la matriz cuyas columnas son las primeras $n+m-2i$ columnas de A_i y N_i es la matriz cuyas columnas son la primeras $n+m-2i-1$ columnas de A_i y la última columna tiene en cada lugar polinomios de grado menor o igual que Y^{i-1}. Observando que $\mathrm{sRes}_i(F,G) = \det(M_i)$, se obtiene $b)$ para $1 \leq i \leq n$. Para $n+1 \leq i \leq m$, el resultado es trivial. □

Observación 1.35 *Para todo $0 \leq i \leq n$, $\mathrm{SRes}_i(F,G)$ es una combinación polinomial de los polinomios F y G, pues al desarrollar el determinante por la última columna se obtiene una expresión de la forma*

$$\sum_{j=0}^{n-i-1} Y^j F(Y)c_j + \sum_{j=0}^{m-i-1} Y^j G(Y)d_j = \Big(\sum_{j=0}^{n-i-1} Y^j c_j\Big)F(Y) + \Big(\sum_{j=0}^{m-i-1} Y^j d_j\Big)G(Y),$$

donde $c_j, d_j \in A$ son un signo por el determinante de una matriz con coeficientes en A.

El siguiente resultado relaciona la $(n-1)$-ésima subresultante entre dos polinomios el resto de una divisíón euclídea:

Lema 1.36 *Sean $F, G \in A[Y]$ de grados m y n respectivamente, $m > n$. Sea SRes_{n-1} la $(n-1)-$ésima subresultante de F y G considerados como polinomios en $A[X][Y]$. Entonces*

$$\mathrm{SRes}_{n-1} = -\mathrm{Resto}((-1)^{\frac{(m-n-1)(m-n)}{2}}\mathrm{cp}(G)^{m-n+1}F, G).$$

Demostración. Si $F(Y) = \sum_{j=0}^{m} a_j Y^j$ y $G(Y) = \sum_{j=0}^{n} b_j Y^j$, SRes_{n-1} es, por definición, el determinante de la matriz de tamaño $m-n+2$:

$$\begin{pmatrix} a_m & a_{m-1} & \cdots & \cdots & a_n & F \\ 0 & 0 & \cdots & \cdots & b_n & G \\ \vdots & & & \iddots & & YG \\ \vdots & & \iddots & \cdots & \cdots & Y^2G \\ 0 & \iddots & \cdots & \cdots & \cdots & \vdots \\ b_n & b_{n-1} & \cdots & \cdots & \cdots & Y^{m-n}G \end{pmatrix}.$$

Desarrolando el determinante por la última columna obtenemos que

$$\mathrm{SRes}_{n-1} = (-1)^{m-n+3}F\det(M) + GH(Y),$$

donde M es la matriz triangular inferior de tamaño $m-n+1$ con b_n en todos los coeficientes de la diagonal y H es un polinomio en $A[Y]$. Utilizando que

$$\det(M) = (-1)^{\frac{(m-n+1)(m-n)}{2}}(b_n)^{m-n+1}$$

y que

$$\mathrm{sg}\big(m-n+3+\frac{(m-n+1)(m-n)}{2}\big) = -\mathrm{sg}\big(\frac{(m-n-1)(m-n)}{2}\big),$$

pues $m-n+3+\frac{(m-n+1)(m-n)}{2} = \frac{(m-n-1)(m-n)}{2} + 2(m-n)+3$, se obtiene que

$$\mathrm{SRes}_{n-1} = -(-1)^{\frac{(m-n-1)(m-n)}{2}} b_n^{m-n+1} F + GH(Y).$$

Si $\mathrm{SRes}_{n-1} \neq 0$, por la Proposición 1.34, tiene grado en Y menor o igual que $n-1$, y se deduce la identidad propuesta. □

Como consecuencia directa del Teorema de estructura para subresultantes (ver [3, Theorem 8.56]), el resultado anterior puede generalizarse a los demás i-ésimos polinomios subresultantes.

Teorema 1.37 *Sean $F, G \in A[Y]$ de grados m y n respectivamente. Supongamos $n < m$. Para $-1 \leq i \leq m$, sea SRes_i la $i-$ésima subresultante de F y G. Entonces para $1 \leq l \leq m$, si SRes_{l-1} es un polinomio no nulo de grado j:*

- *Si $\mathrm{SRes}_{j-1} = 0$, entonces $\mathrm{SRes}_{l-1} = \gcd(F,G)$ salvo un factor en A.*
- *Si $\mathrm{SRes}_{j-1} \neq 0$ tiene grado k,*

$$s_j t_{l-1} \mathrm{SRes}_{k-1} = -\mathrm{Resto}(s_k t_{j-1} \mathrm{SRes}_{l-1}, \mathrm{SRes}_{j-1})$$

y el cociente está en $A[Y]$, donde $s_u := \mathrm{sRes}_u(F,G)$ y t_u es el coeficiente principal de SRes_u.

Es importante observar que la división en el último ítem tiene cociente con coeficientes en el anillo debido a [3, Proposition 8.62].

En el caso particular en que el dominio íntegro sea $\mathbb{Z}[X]$, se obtienen las siguientes cotas para los grados en las variables X e Y y la altura de la $i-$ésima subresultante de F y G y el $i-$ésimo coeficiente subresultante de F y G, vistos como polinomios de variable Y:

Proposición 1.38 *Sean $F, G \in \mathbb{Z}[X,Y]$ polinomios que verifican $\deg_Y(G) < \deg_Y(F)$. Para $i = 0, \ldots, \deg_Y(F)$, consideremos $\mathrm{SRes}_i(F,G)$ la $i-$ésima subresultante de F y G y $\mathrm{sRes}_i(F,G)$ el $i-$ésimo coeficiente subresultante de F y G, vistos como polinomios en $\mathbb{Z}[X][Y]$. Entonces:*

a) $\mathrm{SRes}_i(F,G) \in \mathbb{Z}[X,Y]$ *es el polinomio nulo o bien tiene grado en Y menor o igual que i y tiene grado en X menor o igual que $(\deg_Y(G)-i)\deg_X(F) + (\deg_Y(F)-i)\deg_X(G)$.*

En particular, $\mathrm{sRes}_i(F,G) \in \mathbb{Z}[X]$ es el polinomio nulo o bien tiene grado menor o igual que $(\deg_Y(G)-i)\deg_X(F) + (\deg_Y(F)-i)\deg_X(G)$.

b) Para $i = 0 \ldots, \deg_Y(F)$, la altura de $\mathrm{SRes}_i(F,G)$ está acotada por

$$\big(\deg_Y(F)+\deg_Y(G)-2i\big)!\big(H(F)(\deg_X(F)+1)\big)^{\deg_Y(G)-i}\big(H(G)(\deg_X(G)+1)\big)^{\deg_Y(F)-i}.$$

c) La altura de $\mathrm{Res}_Y(F,G)$ *es menor o igual que*

$$\big(\deg_Y(F)+\deg_Y(G)\big)!\big(H(F)(\deg_X(F)+1)\big)^{\deg_Y(G)}\big(H(G)(\deg_X(G)+1)\big)^{\deg_Y(F)}.$$

Demostración.

a) Si $i=\deg_Y(G)+1,\dots,\deg_Y(F)$, $\mathrm{SRes}_i(F,G)$ es el polinomio nulo, G ó F, por lo que las cotas valen claramente. Si $i=0,\dots,\deg_Y(G)$, por el ítem *a)* de la Proposición 1.34, $\deg_Y(\mathrm{SRes}_i(F,G))\le i$. Respecto al grado en X, $\mathrm{SRes}_i(F,G)$ es el determinante de una matriz que tiene a lo sumo $\deg_Y(F)-i$ filas con coeficientes que son polinomios de grado en X a lo sumo $\deg_X(G)$ y a lo sumo $\deg_Y(G)-i$ filas con coeficientes que son polinomios de grado en X a lo sumo $\deg_X(F)$, de donde se obtiene la cota propuesta. Observando que, por el ítem *b)* de la Proposición 1.34, $\deg_X(\mathrm{sRes}_i(F,G))\le\deg_X(\mathrm{SRes}_i(F,G))$, se obtiene la cota para los coeficientes subresultantes.

b) Llamemos $m=\deg_Y(F)$ y $n=\deg_Y(G)$. Al desarrollar el determinante correspondiente a $\mathrm{SRes}_i(F,G)$ por la última columna se obtiene una expresión de la forma

$$\sum_{j=0}^{n-i-1}Y^jF(X,Y)\beta_j+\sum_{l=0}^{m-i-1}Y^lG(X,Y)\lambda_l,$$

donde β_j es el determinante de una matriz con $n-i-1$ filas que tienen coeficientes de F como polinomio en la variable Y y $m-i$ filas que son coeficientes de G como polinomio de variable Y, para todo $j=1,\dots,n-i-1$, y λ_l es el determinante de una matriz con $n-i$ filas que tienen coeficientes de F como polinomio en la variable Y y $m-i-1$ filas que son coeficientes de G como polinomio en la variable Y, para todo $l=1,\dots,m-i-1$. Luego, por la Proposición 1.11,

$$H(\mathrm{SRes}_i(F,G))\le\sum_{j=0}^{n-i-1}(\deg_X(F)+1)H(F)H(\beta_j)+\sum_{l=0}^{m-i-1}(\deg_X(G)+1)H(G)H(\lambda_l).$$

Como, $H(\beta_j)\le(m+n-2i-1)!\big(H(F)(\deg_X(F)+1)\big)^{n-i-1}\big(H(G)(\deg_X(G)+1)\big)^{m-i}$ y $H(\lambda_l)\le(m+n-2i-1)!\big(H(F)(\deg_X(F)+1)\big)^{n-i}\big(H(G)(\deg_X(G)+1)\big)^{m-i-1}$, para todo $j=1,\dots,n-i-1$ y para todo $l=1,\dots,m-i-1$, se sigue fácilmente la cota propuesta.

c) Es el caso $i=0$ del ítem *b)*, pues por la Observación 1.32, $\pm\mathrm{Res}_Y(F,G)=\mathrm{sRes}_0(F,G)$, que por la Proposición 1.34 *a)*, resulta ser el polinomio $\mathrm{SRes}_0(F,G)$.

□

Observación 1.39 *En particular, si* $F\in\mathbb{Z}[X]$ *y* $G\in\mathbb{Z}[X,Y]$, *con* $\deg(F),\deg_Y(G)>0$, *y* $R(Y):=\mathrm{Res}_X(F(X),G(X,Y))\in\mathbb{Z}[Y]$,

$$H(R)\le\big(\deg(F)+\deg_X(G)\big)!H(F)^{\deg_X(G)}\big((\deg_Y(G)+1)H(G)\big)^{\deg(F)}.$$

Más específicamente, si $G(X,Y)=Y-h(X)$, *con* $h\in\mathbb{Z}[X]$, *se tiene que*

$$H(R)\le\Big(\deg(F)+\deg(h)\Big)!H(F)^{\deg(h)}\big(2H(h)\big)^{\deg(F)}.$$

En lo que sigue y siempre que esté claro a cuáles polinomios F y G nos referimos, escribiremos sRes_i y SRes_i en vez de $\mathrm{sRes}_i(F,G)$ y $\mathrm{SRes}_i(F,G)$ respectivamente, para no recargar la notación.

Corolario 1.40 *Con la misma notación que en el Teorema 1.37, consideremos la secuencia de enteros definida de la siguiente manera:*

$$\begin{cases} n_0 = m+1, \quad n_1 = m \\ n_{i+1} := \deg_Y(\mathrm{SRes}_{n_i-1}) \quad \textit{si } i \geq 1 \textit{ y } \mathrm{SRes}_{n_i-1} \neq 0. \end{cases}$$

Si $\mathrm{SRes}_{n_i-1} = 0$*, finaliza la construcción.*

Notemos $R_i := \mathrm{SRes}_{n_i-1} \in \mathbb{Z}[X][Y]$.

Entonces la secuencia $R_0, R_1, \ldots$ *es finita; es decir, existe* $N := \max\{i \geq 0 \mid R_i \neq 0\}$. *Más aún,* $N \leq n+1$.

Además, si llamamos $\tau_i \in \mathbb{Z}[X]$ *al coeficiente principal de* R_i *para* $i = 0, \ldots, N$, *y* $\rho_i = \mathrm{sRes}_{n_i}(F,G) \in \mathbb{Z}[X]$ *para* $i = 1, \ldots, N+1$, *se tiene que:*

1) *Los grados de* τ_i *y de* ρ_i *están acotados por* $\deg_Y(F)\deg_X(G) + \deg_Y(G)\deg_X(F)$.

2) *Las alturas de los polinomios* τ_i *y* ρ_i *están acotadas por*

$$\big(\deg_Y(F) + \deg_Y(G)\big)!\big(H(F)(\deg_X(F)+1)\big)^{\deg_Y(F)}\big(H(G)(\deg_X(G)+1)\big)^{\deg_Y(G)}.$$

3) *Se verifican las siguientes relaciones, donde* $C_i \in \mathbb{Z}[X,Y]$ *para* $i = 1, \ldots, N-1$*:*

$$(-1)^{(m-n)(m-n-1)/2}\mathrm{cp}(G)^{m-n+1}R_0 = R_1C_1 - R_2 \tag{1.8}$$

$$\rho_{i+2}\tau_{i+1}R_i = R_{i+1}C_{i+1} - \rho_{i+1}\tau_iR_{i+2} \qquad \textit{para } 1 \leq i \leq N-1 \tag{1.9}$$

Demostración. Observemos primero que por la Definición 1.33, $R_0 = \mathrm{SRes}_m = F$ y $R_1 = \mathrm{SRes}_{m-1} = G$. Por la Proposición 1.38, para $i \geq 1$:

$$n_{i+1} = \deg_Y(\mathrm{SRes}_{n_i-1}) \leq n_i - 1 < n_i,$$

y además $n_2 = \deg_Y(\mathrm{SRes}_{n_1-1}) = \deg_Y(\mathrm{SRes}_{m-1}) = \deg_Y(G) = n$. Luego, se obtiene una secuencia estrictamente decreciente de números enteros no negativos

$$0 \leq \ldots < n_3 < n_2 = n < n_1,$$

que necesariamente deberá ser finita. Luego, existe $N \in \mathbb{N}$ tal que $n_{N+1} \geq 0$, $R_N \neq 0$ y $R_{N+1} = 0$. En particular, $N+1 \leq n+2$, o sea, $N \leq n+1$.

1) Por definición, $\tau_i = \mathrm{cp}(R_i) = \mathrm{cp}(\mathrm{SRes}_{n_i-1})$. Luego, por la Proposición 1.38, $\deg(\tau_i) \leq (\deg_Y(G) - n_i + 1)\deg_X(F) + (\deg_Y(F) - n_i + 1)\deg_X(G)$ que es a su vez, menor o igual que $\deg_Y(G)\deg_X(F) + \deg_Y(F)\deg_X(G)$. Por otro lado, como $\rho_i = \mathrm{sRes}_{n_i}$, por la Proposición 1.38, $\deg(\rho_i) \leq (\deg_Y(G) - n_i)\deg_X(F) + (\deg_Y(F) - n_i)\deg_X(G) \leq \deg_Y(G)\deg_X(F) + \deg_Y(F)\deg_X(G)$.

2) Es consecuencia del ítem b) de la Proposición 1.34 y del ítem b) de la Proposición 1.38.

3) El Lema 1.36 dice que existe $C_1 \in \mathbb{Z}[X,Y]$ tal que vale (1.8), y el Teorema 1.37 aplicado a $j = n_i$, para $i = 2, \ldots, N-1$, dice que existen polinomios $C_2, \ldots, C_{N-1} \in \mathbb{Z}[X,Y]$ tales que vale (1.9).

□

Será útil observar las siguientes identidades:

Lema 1.41 *Con las hipótesis y notación del Corolario 1.40:*

a) $\rho_1 = \mathrm{cp}(F)$,

b) $\rho_2 = (-1)^{\frac{(m-n)(m-n-1)}{2}} \mathrm{cp}(G)^{m-n}$,

c) $\tau_0 = \mathrm{cp}(F)$, $\tau_1 = \mathrm{cp}(G)$ *y*

d) $R_N = \gcd(F,G)$ *salvo un factor en* $\mathbb{Z}[X]$.

Si además $\mathrm{Res}_Y(F,G) \in \mathbb{Z}[X]$ *no es cero:*

e) $\rho_{N+1} = \mathrm{Res}_Y(F,G)$ *y*

f) $\tau_N = R_N$

Demostración. Las igualdades de los ítems $a), b)$ y c) salen aplicando la definición. La identidad en d) se obtiene del Teorema 1.37 tomando $l = n_N$ y $j = n_{N+1}$.
Si $\mathrm{Res}_Y(F,G) \neq 0$, F y G no tienen un factor común en $\mathbb{Q}[Y]$. Luego, $\gcd(F,G)$ tiene grado cero en Y y obtenemos (aplicando d)) que $n_{N+1} = \deg_Y(R_N) = 0$. Entonces $\rho_{N+1} = \mathrm{sRes}_{n_{N+1}}(F,G) = \mathrm{sRes}_0(F,G)$ y, por la Observación 1.32, se verifica la identidad propuesta. Por otro lado, como $R_N \in \mathbb{Z}[X]$, $\tau_N = \mathrm{cp}(R_N) = R_N$. □

La construcción definida en el Corolario 1.40 será de vital importancia en los algoritmos que desarrollaremos en el Capítulo 2. Es por eso que introducimos el siguiente resultadosobre la complejidad del cálculo algorítmico de las subresultantes de dos polinomios en dos variables.

Teorema 1.42 *Sean* $F, G \in \mathbb{Z}[X,Y]$ *de grados* m *y* n *en la variable* Y *respectivamente. Supongamos* $n < m$. *Existe un algoritmo que calcula todas las subresultantes de* F *y* G *como polinomios de variable* Y. *Además, si* $\mu \geq n \deg_X(F) + m \deg_X(G)$, *la complejidad de este algoritmo es de orden*

$$O\Big(\mu(\deg_X(F)m + \deg_X(G)n) + n\mu(n+m)^{\omega+1} + n^2 M(\mu)\log(\mu)\Big).$$

Demostración. Por la Proposición 1.38, el grado en X de SRes_i es menor o igual a

$$\deg_Y(G)\deg_X(F) + \deg_Y(F)\deg_X(G) \leq \mu.$$

Para cada $i = 0, \ldots, n$, computamos los coeficientes de $\mathrm{SRes}_i(F,G)$ interpolando en, a lo sumo, $\mu + 1$ puntos de la siguiente manera. Las complejidades de los procedimientos pueden verse en la Sección 1.2.

1) Sean $x_0, \ldots, x_\mu \in \mathbb{Z}$. Evaluemos los coeficientes de F y G en esos puntos. Son $m+1$ polinomios de grado acotado por $\deg_X(F)$ y $n+1$ polinomios de grado acotado por $\deg_X(G)$. Esto puede hacerse con una complejidad de $O\big(\mu(\deg_X(F)m + \deg_X(G)n)\big)$.

2) Para cada $k = 0, \ldots, \mu$, para cada $0 \leq i \leq n$, calculemos el determinante de una matriz de tamaño $(n+m-2i) \times (n+m-2i)$ desarrollándolo por la última columna de la siguiente manera:

   i) Calculemos $n+m-2i$ determinantes de matrices de tamaño $(n+m-2i-1) \times (n+m-2i-1)$. Esto puede hacerse con una complejidad de $O((n+m)^{\omega+1})$.

   ii) Multipliquemos los determinantes anteriores por los polinomios de la forma $Y^t F(x_k, Y)$, para $t \leq n-1$, y $Y^t G(x_k, Y)$, para $t \leq m-1$, y luego sumemos a lo sumo $n+m$ polinomios de grado acotado por $n+m$. Esto puede hacerse con una complejidad de $O((n+m)^2)$.

   Este paso puede realizarse con una complejidad de $O(\mu n(n+m)^{\omega+1})$.

3) Para $i = 0, \ldots, n$, en vista de obtener los coeficientes de $\mathrm{SRes}_i(F, G)$, realicemos $i+1$ interpolaciones en, a lo sumo, $\mu+1$ puntos. Este paso puede hacerse con una complejidad de $O(n^2 M(\mu)\log(\mu))$.

En total, este algoritmo puede hacerse con una complejidad de orden

$$O\Big(\mu(\deg_X(F)m + \deg_X(G)n) + \mu n(n+m)^{\omega+1} + n^2 M(\mu)\log(\mu)\Big).$$

□

Capítulo 2

Algoritmos efectivos para funciones Pfaffianas

En este capítulo nos concentraremos en el desarrollo de algoritmos para resolver ciertos problemas para una clase de funciones de una variable, conocidas como funciones Pfaffianas de orden 1. Comenzaremos, en la Sección 2.1, definiendo estas funciones y analizando algunas de sus propiedades básicas. En la Sección 2.2, generalizaremos la noción de secuencia de Sturm para polinomios, introducida en el Capítulo 1, a estas funciones, lo que nos permitirá generalizar el Teorema 1.4. Daremos una forma algorítmica de calcular esta secuencia y, como consecuencia, propondremos un algoritmo efectivo que calcula la cantidad de ceros en un intervalo cerrado y acotado (con extremos racionales) de una función. En la Sección 2.3, introduciremos la definición de secuencia de Sturm de una función continua respecto a otra y probaremos la relación de ésta con el *indicador de Tarski* de estas funciones. Daremos un algoritmo efectivo para calcular esta secuencia para funciones Pfaffianas de orden 1. En la Sección 2.4, nos centraremos en el problema de decisión para ciertas fórmulas que involucran funciones Pfaffianas de una variable asociadas a una función fija: presentaremos un procedimiento simbólico que resuelve este problema y estimaremos su complejidad.

2.1. Funciones Pfaffianas

Las *funciones Pfaffianas* sobre $\mathbb{R}$, introducidas por Khovanskii a fines de los '70, son funciones analíticas definidas en un abierto $\mathcal{U} \subset \mathbb{R}^n$ que satisfacen sistemas triangulares de ecuaciones diferenciales de primer orden con coeficientes polinomiales.

Más precisamente, sean $f_1, \ldots, f_l$ funciones analíticas definidas en $\mathcal{U}$ para las cuales existen polinomios Φ_{ij} en $n+i$ indeterminadas que satisfacen en $\mathcal{U}$ las ecuaciones

$$\frac{\partial f_i}{\partial x_j}(x) = \Phi_{ij}\Big(x, f_1(x), \ldots, f_i(x)\Big), 1 \leq i \leq l, 1 \leq j \leq n.$$

En este caso, diremos que las funciones $f_1, \ldots, f_l$ forman una *cadena Pfaffiana*. Dado un polinomio $F \in \mathbb{R}[X_1, \ldots, X_n, Y_1, \ldots, Y_l]$, una *función Pfaffiana* definida por F y asociada a la cadena Pfaffiana $f_1, \ldots, f_l$ es una función de la forma

$$f(x) = F(x, f_1(x), \ldots, f_l(x)).$$

Siguiendo [7], diremos que f tiene *orden* l y *grado* $(\delta, \deg(F))$ si δ es una cota superior para el grado de los polinomios Φ_{ij}, con $1 \leq i \leq l, 1 \leq j \leq n$.

Algunos ejemplos de funciones Pfaffianas son los siguientes:

1. Los polinomios son funciones Pfaffianas de orden $l = 0$.
2. Para $F \in \mathbb{R}[X, Y]$ y $f_1(x) = e^x$, la función $f(x) = F(x, f_1(x))$ es una función Pfaffiana de orden $l = 1$ y grado $(1, \deg(F))$ pues $f_1'(x) = f_1(x)$.
3. Para $F \in \mathbb{R}[X, Y, Z]$, $f_1(x) = x^{-1}$ y $f_2(x) = \ln(x)$, la función $f(x) = F(x, f_1(x), f_2(x))$ es una función Pfaffiana en $\mathbb{R}_{\geq 0}$ de orden $l = 2$ y grado $(2, \deg(F))$ pues $f_1'(x) = -f_1^2(x)$ y $f_2'(x) = f_1(x)$.
4. Para $F \in \mathbb{R}[X, Y, Z]$ y $f_1(x, y) = e^{x-y}$, la función $f(x, y) = F(x, y, f_1(x, y))$ es una función Pfaffiana en $\mathbb{R}^2$ de orden $l = 1$ y grado $(1, \deg(F))$ pues $\frac{\partial f_1}{\partial x}(x, y) = f_1(x, y)$ y $\frac{\partial f_1}{\partial y}(x, y) = -f_1(x, y)$.

El Teorema de Khovanskii establece que un sistema de n ecuaciones Pfaffianas en n variables definidas en un dominio $\mathcal{U}$ tiene finitas soluciones reales no degeneradas en $\mathcal{U}$ y que la cantidad de estas soluciones puede acotarse explícitamente en términos de parámetros sintácticos asociados al sistema:

Teorema 2.1 *([7, Theorem 3.1]) Sea $f_1, \ldots, f_r$ una cadena Pfaffiana definida en un dominio $\mathcal{U} \subset \mathbb{R}^n$ y sean $g_1, \ldots, g_n$ funciones Pfaffianas de grado (δ, d_i), para $i = 1, \ldots, n$, asociadas a la cadena Pfaffiana anterior. La cantidad de soluciones no degeneradas del sistema*

$$g_1(x) = 0, \ldots, g_n(x) = 0$$

es menor o igual que

$$2^{\frac{r}{2}(r-1)} \prod_{1 \leq i \leq n} d_i \Big(\min\{n, r\}\, \delta - n + 1 + \sum_{1 \leq i \leq n} d_i \Big)^r.$$

En esta tesis trabajaremos con funciones Pfaffianas de una variable de orden 1 definidas por polinomios en $\mathbb{Z}[X, Y]$, es decir, para el caso $n = 1$ y $l = 1$. Por esta razón, reformularemos la definición de manera más sencilla.

Definición 2.2 *Sea φ una función analítica definida en un abierto $\mathcal{U} \subset \mathbb{R}$ para la cual existe un polinomio $\Phi \in \mathbb{Z}[X, Y]$, con $\deg_Y(\Phi) > 0$, tal que φ satisface en $\mathcal{U}$ la ecuación*

$$\varphi'(x) = \Phi(x, \varphi(x)).$$

Dado un polinomio $F \in \mathbb{Z}[X, Y]$ de grado total d, diremos que la función

$$f(x) = F(x, \varphi(x))$$

es una función Pfaffiana asociada a *φ definida en $\mathcal{U} \subset \mathbb{R}$, de orden 1 y grado $(\deg(\Phi), d)$. Usaremos la notación d_X, d_Y para referirnos al grado en la variable X y al grado en la variable Y de F respectivamente.*

A lo largo de este trabajo supondremos que la función φ está fija, por lo que será útil fijar la notación δ_X, δ_Y para referirnos a los grados en X y en Y de Φ respectivamente y δ para el grado total de Φ.

La clase de las funciones Pfaffianas asociadas a la función φ y definidas por polinomios en $\mathbb{Z}[X,Y]$ es claramente cerrada para la suma y el producto. Más aún, es cerrada para la derivación:

Lema 2.3 *Dada una función Pfaffiana $f(x) = F(x, \varphi(x))$ con $F \in \mathbb{R}[X,Y]$, llamaremos $\widetilde{F} \in \mathbb{Z}[X,Y]$ al polinomio*

$$\widetilde{F}(X,Y) = \frac{\partial F}{\partial X}(X,Y) + \frac{\partial F}{\partial Y}(X,Y)\Phi(X,Y). \tag{2.1}$$

Entonces $f'(x) = \widetilde{F}(x, \varphi(x))$, por lo que resulta ser una función Pfaffiana asociada a φ. Además, si d_X, d_Y son los grados de F en la variable X e Y respectivamente,

$$\begin{cases} \deg_X(\widetilde{F}) \leq d_X + \delta_X \\ d_Y \leq \deg_Y(\widetilde{F}) \leq d_Y + \delta_Y - 1 \end{cases} \tag{2.2}$$

y, para todo $k \in \mathbb{N}$, la derivada $k-$ésima de f es una función Pfaffiana asociada a φ definida por un polinomio $F_k \in \mathbb{Z}[X,Y]$ que verifica

$$\begin{cases} \deg_X(F_k) \leq d_X + k\delta_X \\ \deg_Y(F_k) \leq d_Y + k(\delta_Y - 1). \end{cases}$$

Demostración. Para demostrar (2.2), solo basta probar que $d_Y \leq \deg_Y(\widetilde{F})$.

Sabemos que $\deg_Y(\frac{\partial F}{\partial Y}\Phi) = d_Y - 1 + \delta_Y$. Se tienen dos casos:

- Si $\delta_Y \geq 2$: Se tiene que $\deg_Y(\frac{\partial F}{\partial Y}\Phi) \geq d_Y + 1 > d_Y \geq \deg_Y(\frac{\partial F}{\partial X})$. Luego, $\deg_Y(\widetilde{F}) = \deg_Y(\frac{\partial F}{\partial Y}\Phi) > d_Y$.
- Si $\delta_Y = 1$: Supongamos que $\Phi(X,Y) = \Phi_0(X) + \Phi_1(X)Y$, $\Phi_1 \neq 0$, y que $F(X,Y) = \sum_{i=0}^{d_Y} a_i(X)Y^i$, con $a_{d_Y} \neq 0$. Se tiene que

$$\widetilde{F}(X,Y) = (a'_{d_Y}(X) + d_Y a_{d_Y}(X)\Phi_1(X))Y^{d_Y} + \sum_{i=0}^{d_Y-1} b_i(X)Y^i,$$

  con $b_i \in \mathbb{Z}[X]$, $i = 0, \ldots, d_Y - 1$. Como $\deg_X(a'_{d_Y}) < \deg_X(a_{d_Y})$, se tiene que $a'_{d_Y}(x) + d_Y a_{d_Y}(X)\Phi_1(X) \neq 0$ y, por lo tanto, $\deg_Y(\widetilde{F}) = d_Y$.

Para probar las cotas para los grados de F_k, sólo hay que observar que $F_1 = \widetilde{F}$, que $F_k = \widetilde{F_{k-1}}$ y usar inducción. □

Para estimar la complejidad de los algoritmos que desarrollaremos, necesitaremos una cota superior para la multiplicidad de un cero de una función Pfaffiana asociada a φ.

Recordemos que si f es una función analítica en un dominio $\mathcal{U} \subset \mathbb{R}$ y $\alpha \in \mathcal{U}$ es un cero de f, la *multiplicidad de α como cero de f* se define como

$$\text{mult}(\alpha, f) = \text{mín}\left\{ \; r \in \mathbb{N}_0 / \; f^{(r)}(\alpha) \neq 0 \; \right\},$$

admitiendo las notaciones $f^{(0)}$ para la función f sin derivar y $\text{mult}(\alpha, f) = 0$ para un α que no es cero de f.

A continuación, daremos una cota que dependerá de cada uno de los grados en las variables X e Y de los polinomios que definen estas funciones. Una cota superior dependiendo del grado total de los polinomios se puede hallar en [7, Theorem 4.3]. Aunque ambas cotas sean del mismo orden, nuestra cota puede ser menor cuando el grado total es mayor que los grados con respecto a cada variable.

Proposición 2.4 *Sea $f(x) = F(x, \varphi(x))$ una función Pfaffiana no nula asociada a φ, con $F \in \mathbb{Z}[X, Y]$ y $\deg_Y(F) > 0$. Si $\alpha \in \mathbb{R}$ y $f(\alpha) = 0$ entonces*

$$\text{mult}(\alpha, f) \leq 2 \deg_X(F) \deg_Y(F) + \deg_X(F)(\delta_Y - 1) + \deg_Y(F)(\delta_X + 1).$$

Demostración. Supongamos primero que F es irreducible en $\mathbb{Q}[X, Y]$.
Si $f(\alpha) = 0$, luego $\text{mult}(\alpha, f) > \text{mult}(\alpha, f')$. Como $f'(x) = \widetilde{F}(x, \varphi(x))$, luego F no divide a $\widetilde{F}$ y, por lo tanto, $R := \text{Res}_Y(F, \widetilde{F}) \neq 0$. Sean $S, T \in \mathbb{Z}[X, Y]$ tales que $R = SF + T\widetilde{F}$. Se obtiene que

$$R(x) = S(x, \varphi(x)).\, f(x) + T(x, \varphi(x)).\, f'(x).$$

Si α es una raíz múltiple de f, la igualdad anterior implica que $\text{mult}(\alpha, f) \leq \text{mult}(\alpha, R) + 1 \leq \deg(R) + 1$. Teniendo en cuenta que $\deg(R) \leq \deg_X(F) \deg_Y(\widetilde{F}) + \deg_X(\widetilde{F}) \deg_Y(F)$ (ver Proposición 1.30), $\deg_X(\widetilde{F}) \leq \deg_X(F) + \delta_X$ y $\deg_Y(\widetilde{F}) \leq \deg_Y(F) - 1 + \delta_Y$, se concluye que

$$\text{mult}(\alpha, f) \leq 2 \deg_X(F) \deg_Y(F) + \deg_X(F)(\delta_Y - 1) + \delta_X \deg_Y(F) + 1.$$

Para el caso general, sea $F = c(X) \prod_{1 \leq i \leq t} F_i(X, Y)^{m_i}$, donde $c(X) = \text{cont}(F)$ es el contenido de F como polinomio de variable Y y $F_1, \ldots, F_t \in \mathbb{Z}[X, Y]$ son polinomios irreducibles en $\mathbb{Q}[X, Y]$. Para todo i, sea $f_i(x) = F_i(x, \varphi(x))$. De la cota anterior, se deduce que

$$\text{mult}(\alpha, f) = \text{mult}(\alpha, c) + \sum_{1 \leq i \leq t} m_i \, \text{mult}(\alpha, f_i) \leq$$

$$\leq \deg_X(c) + \sum_{1 \leq i \leq t} m_i \left(2 \deg_X(F_i) \deg_Y(F_i) + \deg_X(F_i)(\delta_Y - 1) + \delta_X \deg_Y(F_i) + 1\right).$$

Como $\deg_X(c) + \sum_{1 \leq i \leq t} m_i \deg_X(F_i) \leq \deg_X(F)$, $\sum_{1 \leq i \leq t} m_i \deg_Y(F_i) = \deg_Y(F)$ y $\sum_{1 \leq i \leq t} m_i \leq \deg_Y(F)$, obtenemos que

$$\text{mult}(\alpha, f) \leq 2 \deg_X(F) \deg_Y(F) + \deg_X(F)(\delta_Y - 1) + (\delta_X + 1) \deg_Y(F).$$

□

2.2. Secuencias de Sturm y conteo de ceros

En esta sección, comenzaremos extendiendo la noción de secuencia de Sturm de polinomios de una variable a funciones continuas de una variable y veremos cómo esto permite generalizar el Teorema 1.4 a funciones continuas. Luego, daremos una forma constructiva de obtener una secuencia de Sturm para una función Pfaffiana asociada a φ y deduciremos una cota superior para la cantidad de ceros de una función Pfaffiana del tipo considerado en un intervalo acotado contenido en su dominio. Finalmente, propondremos, dada una función Pfaffiana asociada a φ, un algoritmo para calcular la cantidad de ceros que ésta tiene en un intervalo cerrado y acotado (con extremos racionales), incluido en el dominio de φ, suponiendo, como es usual en la bibliografía, que se dispone de un oráculo para conocer su signo en números reales algebraicos dados por su codificación de Thom como ceros de un polinomio.

2.2.1. Secuencia de Sturm para funciones continuas

Siguiendo [9], introduciremos la noción de secuencia de Sturm para funciones continuas en un intervalo real, que nos permitirá generalizar resultados conocidos para polinomios de una variable (ver Definición 1.2 y Teorema 1.4).

Definición 2.5 *Sea $f_0 : (a, b) \to \mathbb{R}$ una función continua. Una secuencia finita de funciones continuas $\mathbf{f} = (f_0, \ldots, f_N)$ sobre (a, b) se llama* secuencia de Sturm para f_0 en el intervalo (a, b) *si se cumplen las siguientes condiciones:*

1. *Si $f_0(y) = 0$, existe $\epsilon > 0$ tal que $f_1(x) \neq 0$ para todo $x \in (y - \epsilon, y + \epsilon) \subseteq (a, b)$ y*

$$\begin{cases} f_0(x)f_1(x) < 0 & \text{para } \ y - \epsilon < x < y \\ f_0(x)f_1(x) > 0 & \text{para } \ y < x < y + \epsilon. \end{cases}$$

2. *Para todo $i = 1, \ldots, N - 1$, si $f_i(y) = 0$ para $y \in (a, b)$ entonces, $f_{i-1}(y)f_{i+1}(y) < 0$.*

3. *$f_N(y) \neq 0$ para todo $y \in (a, b)$.*

Extendiendo la notación del Teorema 1.4, dada $\mathbf{f} = (f_0, \ldots, f_N)$ una secuencia de funciones definidas en (a, b), para $x \in (a, b)$, $v(\mathbf{f}, x)$ denotará la cantidad de cambios de signo de la $(N+1)$-upla $(f_0(x), \ldots, f_N(x))$. Obtenemos entonces, un resultado análogo al Teorema clásico de Sturm para polinomios. Su demostración puede hallarse en [9, Theorem 2.1].

Teorema 2.6 *Sea $f_0 : (a, b) \to \mathbb{R}$ una función continua. Sea $\mathbf{f} = (f_0, \ldots, f_N)$ una secuencia de Sturm para f_0 en el intervalo (a, b) y sean $a < c < d < b$. Entonces,*

$$\#\{\ x \in (c, d] \ /\ f_0(x) = 0\ \} = v(\mathbf{f}, c) - v(\mathbf{f}, d).$$

2.2.2. Construcción de una secuencia de Sturm

Con el objeto de determinar la cantidad de ceros que la función tiene en un intervalo, en esta sección veremos cómo construir secuencias de Sturm para una función Pfaffiana.

Sea f una función Pfaffiana asociada a φ, verificando las hipótesis de la Definición 2.2. Consideremos el polinomio

$$\widetilde{F}(X,Y) = \frac{\partial F}{\partial X}(X,Y) + \frac{\partial F}{\partial Y}(X,Y)\Phi(X,Y),$$

definido en (2.1).

Dado que buscamos calcular la cantidad de ceros reales de la función f, podremos suponer sin pérdida de generalidad que $\mathrm{Res}_Y(F,\widetilde{F}) \neq 0$ debido al siguiente resultado:

Lema 2.7 *Dada una función Pfaffiana $f(x) = F(x,\varphi(x))$ con las hipótesis anteriores y tal que $\deg_Y(F) > 0$, existe un polinomio $P \in \mathbb{Z}[X,Y]$ tal que $\mathrm{Res}_Y(P,\widetilde{P}) \neq 0$ y $P(x,\varphi(x))$ tiene los mismos ceros reales que $f(x)$. Más aún, el polinomio P puede ser efectivamente calculado a partir de F y de Φ.*

Demostración. Sin pérdida de generalidad, podemos asumir que F es libre de cuadrados. Supongamos que $\mathrm{Res}_Y(F,\widetilde{F}) = 0$. Sea $F_0 \in \mathbb{Z}[X,Y]$ tal que $F = \mathrm{cont}(F)F_0$, donde $\mathrm{cont}(F) \in \mathbb{Z}[X]$ es el contenido de F como polinomio de variable Y. Luego, $\mathrm{Res}_Y(F_0,\widetilde{F}_0) = 0$, y $\gcd(F_0,\widetilde{F}_0)$ es un polinomio $S \in \mathbb{Z}[X,Y]$ de grado positivo en Y. Si

$$F_0 = S\,U \quad \text{y} \quad \widetilde{F}_0 = S\,V$$

para $U, V \in \mathbb{Z}[X,Y]$, se tiene que

$$f_0(x) = F_0(x,\varphi(x)) = S(x,\varphi(x))\,U(x,\varphi(x)) \;\text{ y }\; f_0'(x) = \widetilde{F}_0(x,\varphi(x)) = S(x,\varphi(x))\,V(x,\varphi(x)),$$

lo cual implica que un cero ξ de f_0 que no sea cero de $U(x,\varphi(x))$ satisface que

$$\mathrm{mult}(\xi,f_0) = \mathrm{mult}(\xi,S(x,\varphi(x))) \leq \mathrm{mult}(\xi,f_0'),$$

llegando a una contradicción. Luego, f_0 y $U(x,\varphi(x))$ tienen el mismo conjunto de ceros en $\mathbb{R}$. Como

$$\widetilde{F}_0 = \widetilde{(S\,U)} = \widetilde{S}\,U + S\,\widetilde{U},$$

se sigue que, si $T \in \mathbb{Z}[X,Y]$ es un factor común de U y $\widetilde{U}$ con grado positivo en Y, T divide a $\widetilde{F}_0 = S\,V$. Dado que U y V son polinomios coprimos, se tiene que T divide a S y, por lo tanto, T^2 divide a F_0, contradiciendo el hecho de que F_0 es libre de cuadrados.

Luego, alcanza con tomar $P = \mathrm{cont}(F)\,U$. □

Supongamos entonces que $\mathrm{Res}_Y(F,\widetilde{F}) \neq 0$. En lo que sigue, aplicaremos la teoría de subresultantes introducida en el Capítulo 1 para obtener una secuencia de Sturm para f en un intervalo I contenido en su dominio, bajo ciertas hipótesis adicionales sobre I.

Sea

$$F_1 = \mathrm{Resto}\Big(\mathrm{cp}(F)^D\widetilde{F}, F\Big) \in \mathbb{Z}[X][Y],$$

donde D es el menor entero par mayor o igual que $1 + \deg_Y(\widetilde{F}) - \deg_Y(F)$.

Siguiendo la construcción del Corolario 1.40 aplicado a F y a F_1 se introduce la siguiente notación:

Notación 2.8 *Definimos:*

- $n_0 := \deg_Y(F) + 1, \quad R_0 := F$;
- $n_1 := \deg_Y(F), \quad R_1 := F_1$;
- *para* $i \geq 1$, *si* $R_i \neq 0$,

$$n_{i+1} := \deg_Y(R_i) \text{ y } R_{i+1} := \mathrm{SRes}_{n_{i+1}-1}(F, F_1) \in \mathbb{Z}[X][Y].$$

Sea $N := \max\{i \geq 0 \mid R_i \neq 0\}$. *Para* $i = 0, \ldots, N$, *notamos* $\tau_i \in \mathbb{Z}[X]$ *el coeficiente principal de* R_i *y, para* $i = 2, \ldots, N+1$, *notamos* $\rho_i = \mathrm{sRes}_{n_i}(F, F_1) \in \mathbb{Z}[X]$.

Como consecuencia del Corolario 1.40, $N \leq \deg_Y(F)$ y se obtienen las siguientes relaciones:

$$\beta \ \mathrm{cp}(F_1)^{\deg_Y(F)-\deg_Y(F_1)+1} R_0 = R_1 C_1 - R_2 \tag{2.3}$$

$$\rho_{i+2}\tau_{i+1} R_i = R_{i+1} C_{i+1} - \rho_{i+1}\tau_i R_{i+2} \qquad \text{para } i \geq 1 \tag{2.4}$$

donde $\beta := (-1)^{(\deg_Y(F)-\deg_Y(F_1))(\deg_Y(F)-\deg_Y(F_1)-1)/2}$ y $C_i \in \mathbb{Z}[X][Y]$ para todo i.

Definición 2.9 *Dado un intervalo* $I \subset \mathrm{Dom}(\varphi)$ *que no contenga raíces de* τ_i *para* $i = 0, \ldots, N$ *ni de* ρ_i *para* $i = 2, \ldots, N+1$, *se define inductivamente una secuencia de signos* $(\sigma_{I,i})_{0 \leq i \leq N} \in \{1, -1\}^{N+1}$ *de la siguiente manera:*

- $\sigma_{I,0} = \sigma_{I,1} = 1$,
- $\sigma_{I,2} = \beta \ \mathrm{sg}_I(\mathrm{cp}(F_1))^{\deg_Y(F)-\deg_Y(F_1)+1}$,
- $\sigma_{I,i+2} = \mathrm{sg}_I(\rho_{i+2}\tau_{i+1}\rho_{i+1}\tau_i)\sigma_{I,i}$, *si* $2 \leq i \leq N-2$,

donde, para una función g *continua de una variable sin ceros en* I, $\mathrm{sg}_I(g)$ *denota el signo (constante) de* g *en* I.

Definimos además, para $i = 0, \ldots, N$, *los polinomios*

$$F_{I,i} = \sigma_{I,i} R_i \in \mathbb{Z}[X, Y]$$

e introducimos la secuencia de funciones Pfaffianas $\mathbf{f}_I = (f_{I,i})_{0 \leq i \leq N}$ *definidas por*

$$f_{I,i}(x) = F_{I,i}(x, \varphi(x)).$$

Proposición 2.10 *Con las hipótesis y notaciones anteriores, la secuencia de funciones Pfaffianas* $\mathbf{f}_I = (f_{I,i})_{0 \leq i \leq N}$ *es una secuencia de Sturm para* f *en* $I = (a, b)$ *(ver Definición 2.5).*

Demostración. Para simplificar, el intervalo I estará fijo y así podremos omitir los subíndices I en toda la demostración.

Primero probemos que f_0 y f_1 no tienen ceros en común en I. Supongamos que $\alpha \in I$ lo sea. Luego $F(\alpha, \varphi(\alpha)) = 0$ y $F_1(\alpha, \varphi(\alpha)) = 0$; por lo tanto (por el Lema 1.41), $\rho_{N+1}(\alpha) = \mathrm{Res}_Y(F, F_1)(\alpha) = 0$, lo cual es una contradicción. Teniendo en cuenta además que $f_0 = f$ y que τ_0, el coeficiente principal de F, no se anula en I (y por lo tanto, f_1 tiene el mismo signo que f' en cualquier cero de f en I), se verifica la condición 1 de la Definición 2.5.

Para probar la condición 2, primero notemos que si $f_j(\alpha) = 0$ y $f_{j+1}(\alpha) = 0$ para algún $\alpha \in I$, como ρ_i y τ_i no tienen ceros en I, para todo i, y valen las identidades (2.3) y (2.4), α es un cero común de todas las f_i's, contracidiendo el hecho de que f_0 y f_1 no tienen ceros en común en I. Luego, la condición 2 de la Definición 2.5 se sigue por la definición de los signos $(\sigma_i)_{0 \le i \le N}$ y las identidades (2.3) y (2.4).

La condición 3 se verifica pues, por el Lema 1.41, τ_N es f_N salvo un signo; y τ_N no se anula en I por hipótesis. □

En vista de poder trabajar en un intervalo que contenga raíces de los polinomios de la Definición 2.9, introducimos la idea del signo de una función analítica f en un entorno a la izquierda o a la derecha de un punto del dominio de f. Más formalmente, introducimos la siguiente definición:

Definición 2.11 *Sea $f : J \to \mathbb{R}$ una función analítica no nula definida en un intervalo abierto $J \subset \mathbb{R}$ y sea $c \in J$. Llamemos $\mathrm{sg}(f, c^+)$ al signo que f toma en $(c, c+\varepsilon)$ y $\mathrm{sg}(f, c^-)$ al signo que f toma en $(c-\varepsilon, c)$ para $\varepsilon > 0$ suficientemente chico.*

Para una secuencia de funciones analíticas no nulas $\mathbf{f} = (f_0, \ldots, f_N)$ definida en J, notaremos $v(\mathbf{f}, c^+)$ a la cantidad de cambios de signo de la secuencia $(\mathrm{sg}(f_0, c^+), \ldots, \mathrm{sg}(f_N, c^+))$ y $v(\mathbf{f}, c^-)$ a la cantidad de cambios de signo de la secuencia $(\mathrm{sg}(f_0, c^-), \ldots, \mathrm{sg}(f_N, c^-))$.

Los signos $\mathrm{sg}(f, c^+)$ y $\mathrm{sg}(f, c^-)$ pueden obtenerse a partir de los signos de las derivadas sucesivas de f en c:

Observación 2.12 *Sean $f : J \to \mathbb{R}$ una función analítica no nula definida en un intervalo abierto $J \subset \mathbb{R}$ y $c \in J$. Si $\mathrm{mult}(c, f) = r$, se tiene que*

$$\mathrm{sg}(f, c^+) = \mathrm{sg}\big(f^{(r)}(c)\big) \quad y \quad \mathrm{sg}(f, c^-) = \mathrm{sg}\big((-1)^r f^{(r)}(c)\big).$$

Proposición 2.13 *Con las hipótesis y notación de la Proposición 2.10, si suponemos además que el intervalo cerrado $[a, b]$ está contenido en el dominio de φ entonces*

$$\#\{x \in (a, b) \ / \ f(x) = 0\} = v(\mathbf{f}_I, a^+) - v(\mathbf{f}_I, b^-).$$

Demostración. Es consecuencia de la Proposición 2.10 y del Teorema 2.6. □

Como consecuencia, obtenemos una fórmula para la cantidad de ceros de una función Pfaffiana f de orden 1 en cualquier intervalo acotado I, con su clausura $\bar{I}$ incluida en el dominio de f:

Teorema 2.14 *Sea φ una función Pfaffiana definida en un dominio $\mathcal{U} \subset \mathbb{R}$ y que satisface la ecuación $\varphi'(x) = \Phi(x, \varphi(x))$ en $\mathcal{U}$, para un polinomio $\Phi \in \mathbb{Z}[X, Y]$ con $\deg_Y(\Phi) > 0$. Sea $f(x) = F(x, \varphi(x))$, donde $F \in \mathbb{Z}[X, Y]$ y $\deg_Y(F) > 0$. Supongamos que $\mathrm{Res}_Y(F, \widetilde{F}) \neq 0$ y que $(a, b) \subset \mathbb{R}$ es un intervalo abierto y acotado tal que $[a, b]$ está contenido en $\mathcal{U}$.*

Sean además, ρ_i y τ_i los polinomios en $\mathbb{Z}[X]$ introducidos en la Notación 2.8.
Si $\alpha_1 < \alpha_2 < \ldots < \alpha_k$ son todas las raíces en (a, b) de ρ_i, para $2 \leq i \leq N + 1$ y τ_i, para $0 \leq i \leq N$, entonces

$$\#\{x \in [a, b] \ / \ f(x) = 0\} = \#\{0 \leq j \leq k + 1 \ / \ f(\alpha_j) = 0\} + \sum_{j=0}^{k} v(\mathbf{f}_{I_j}, \alpha_j^+) - v(\mathbf{f}_{I_j}, \alpha_{j+1}^-) \ ,$$

donde $\alpha_0 = a$, $\alpha_{k+1} = b$ y, para todo $0 \leq j \leq k$, $I_j = (\alpha_j, \alpha_{j+1})$ y $\mathbf{f}_{I_j}$ es la secuencia de funciones introducidas en la Definición 2.9.

El Teorema de Khovanskii (ver Teorema 2.1) da una cota superior para la cantidad de ceros no degenerados de una función Pfaffiana de una variable en un intervalo abierto. Más aún, puede obtenerse una cota para la multiplicidad de uno de sus ceros (ver [7, Theorem 4.3]).

Dado un polinomio $F \in \mathbb{Z}[X, Y]$, $\deg(F) = d$ que define una función Pfaffiana $f(x) = F(x, \varphi(x))$ asociada a φ, usando las cotas de este teorema (para $n = r = 1$), se sigue que tanto la cantidad de ceros no degenerados en un intervalo abierto como la multiplicidad de un cero arbitrario de f son a lo sumo $d(\delta + d)$ (recordemos que $\delta = \deg(\Phi)$). Se obtiene entonces una cota para la cantidad total de ceros de f en un intervalo abierto acotando el número de ceros no degenerados de f y de sus derivadas sucesivas de orden a lo sumo $d(\delta + d) - 1$.

Por el Lema 2.3, para todo $k \in \mathbb{N}$, $f^{(k)}$ está definida por un polinomio de grado a lo sumo $d + k(\delta - 1)$. Luego, la cantidad total de ceros de f es a lo sumo

$$\sum_{k=0}^{d(\delta+d)-1} (d + k(\delta - 1))(\delta + d + k(\delta - 1)) \leq \frac{1}{2} d^3 \delta^2 (\delta + d)^3. \tag{2.5}$$

Una cota superior alternativa, de menor orden que ésta, puede obtenerse del Teorema 2.14:

Corolario 2.15 *Sea φ una función definida en un abierto $\mathcal{U} \subset \mathbb{R}$ y que satisface la ecuación $\varphi'(x) = \Phi(x, \varphi(x))$ en $\mathcal{U}$, para un polinomio $\Phi \in \mathbb{Z}[X, Y]$ con $\deg_Y(\Phi) > 0$. Sea $f(x) = F(x, \varphi(x))$, con $F \in \mathbb{Z}[X, Y]$ y grado $d_Y > 0$ en Y. Supongamos que $\mathrm{Res}_Y(F, \widetilde{F}) \neq 0$ y que $J \subset \mathcal{U}$ es un intervalo abierto y acotado. Entonces*

$$\#\{\ x \in J \ / \ f(x) = 0\ \} \leq (d_Y + 1)(2d_Y^2 - d_Y)\big((\delta_Y + 3)d_X + \delta_X\big) + d_Y,$$

donde $d_X := \deg_X(F)$, $\delta_X := \deg_X(\Phi)$ y $\delta_Y := \deg_Y(\Phi)$.

Demostración. Sabemos, por (2.5), que la cantidad total de ceros de f en J es finita. Luego, existe un intervalo I abierto y acotado tal que $\bar{I} \subset J \subset \mathcal{U}$ y

$$\#\{\ x \in J \ / \ f(x) = 0\ \} = \#\{\ x \in I \ / \ f(x) = 0\ \}.$$

A su vez, teniendo en cuenta que los extremos del intervalo I no son ceros de f, por el Teorema 2.14 aplicado al intervalo I se obtiene que

$$\#\{\ x \in I\ /\ f(x)=0\ \} = \#\{\ 1 \le j \le k\ /\ f(\alpha_j)=0\ \} + \sum_{j=0}^{k} v(\mathbf{f}_{I_j}, \alpha_j^+) - v(\mathbf{f}_{I_j}, \alpha_{j+1}^-) \le$$

$$\le k + (k+1)N = k(N+1) + N,$$

donde N es la longitud de la secuencia $\mathbf{f}$ definida en la Notación 2.8.

Veamos cómo acotar k y N.

Por el Corolario 1.40, $N \le \deg_Y(F_1) + 1 \le d_Y$ y cada polinomio ρ_i, con $3 \le i \le N+1$ y τ_i, con $2 \le i \le N$, tiene grado acotado por

$$\deg_X(F_1)d_Y + d_X \deg_Y(F_1).$$

Además, el Lema 1.41 afirma que $\deg(\tau_0) \le d_X$, $\deg(\tau_1) \le \deg_X(F_1)$ y ρ_2 tiene los mismos ceros que τ_1. Esto dice que

$$k \le (2d_Y - 2)\big(\deg_X(F_1)\deg_Y(F) + \deg_X(F)\deg_Y(F_1)\big) + \deg(\tau_0) + \deg(\tau_1) \le$$

$$\le (2d_Y - 2)\big(d_Y \deg_X(F_1) + d_X(d_Y - 1)\big) + d_X + \deg_X(F_1). \tag{2.6}$$

Veamos que $\deg_X(F_1) \le (\delta_Y + 2)d_X + \delta_X$.

El polinomio F_1 se obtiene mediante el algoritmo de división (respecto a la variable Y) en a lo sumo D pasos, donde D es el menor entero par mayor o igual que $1 + deg_Y(\widetilde{F}) - d_Y$. Si llamamos $r_0 := \mathrm{cp}(F)^D \widetilde{F}$, veamos por inducción en i que, en el paso $i \ge 1$ de la división, se obtiene un polinomio no nulo $r_i \in \mathbb{Z}[X,Y]$ con $r_i = \mathrm{cp}(F)^{D-i} L_i(X,Y)$, donde $L_i \in \mathbb{Z}[X][Y]$ y $\deg_X(L_i) \le (i+1)d_X + \delta_X$:

- Si $i = 1$, se tiene que $r_0 = \mathrm{cp}(F)^{D-1}\mathrm{cp}(\widetilde{F})Y^{\deg_Y(\widetilde{F})-d_Y}F + r_1$. Observar que $r_1 \neq 0$ pues $\mathrm{Res}_Y(F,\widetilde{F}) \neq 0$. Si definimos $L_1(X,Y) := \mathrm{cp}(F)\widetilde{F} - \mathrm{cp}(\widetilde{F})Y^{\deg_Y(\widetilde{F})-d_Y}F$, obtenemos que $r_1 = \mathrm{cp}(F)^{D-1}L_1$, con $\deg_X(L_1) \le d_X + \deg_X(\widetilde{F}) \le 2d_X + \delta_X$ (ver (2.2)).

- Supongamos que la afirmación vale para el paso $i \ge 1$. Por el algoritmo de división, r_{i+1} es el polinomio que verifica que

  $$r_i = \mathrm{cp}(F)^{D-i-1}\mathrm{cp}(L_i)Y^{\deg_Y(L_i)-d_Y}F + r_{i+1},$$

  con $\deg_Y(r_{i+1}) < \deg_Y(r_i)$ (observar que r_{i+1} es no nulo pues $\mathrm{Res}_Y(F,\widetilde{F}) \neq 0$). Como $r_i = \mathrm{cp}(F)^{D-i}L_i(X,Y)$ por hipótesis inductiva, si llamamos $L_{i+1}(X,Y) := \mathrm{cp}(F)(X)L_i(X,Y) - \mathrm{cp}(L_i)(X)F(X,Y)Y^{\deg_Y(L_i)-d_Y}$, se tiene que

  $$r_{i+1} = \mathrm{cp}(F)^{D-i-1}L_{i+1},$$

  donde $\deg_X(L_{i+1}) \le d_X + \deg_X(L_i) \le (i+2)d_X + \delta_X$.

Sea $i_0 = \mín\{i\ /\ \deg_Y(r_i) < d_Y\} \le \delta_Y$, se tiene que

$$F_1 = r_{i_0} = \mathrm{cp}(F)^{D-i_0}L_{i_0}$$

y, por lo tanto, $\deg_X(F_1) \leq (D - i_0)d_X + (i_0 + 1)d_X + \delta_X \leq (D + 1)d_X + \delta_X$. Dado que $D \leq \delta_Y + 1$ (por (2.2)), se tiene que $\deg_X(F_1) \leq (\delta_Y + 2)d_X + \delta_X$.

De esta forma obtenemos que k puede acotarse por

$$(2d_Y - 2)\big((\delta_Y + 3)d_X d_Y + \delta_X d_Y - d_X\big) + d_X + (\delta_Y + 2)d_X + \delta_X =$$

$$(2d_Y - 2)\left[(\delta_Y + 3)d_X d_Y + d_Y\delta_X - d_X\right] + (\delta_Y + 3)d_X + \delta_X =$$

$$= (2d_Y^2 - 2d_Y + 1)\left[(\delta_Y + 3)d_X + \delta_X\right] - (2d_Y - 2)d_X.$$

Luego,

$$k(N+1) + N \leq \left[(\delta_Y + 3)d_X + \delta_X\right](d_Y + 1)\big(2d_Y^2 - d_Y - (d_Y - 1)\big) - 2(d_Y - 1)d_X + d_Y \leq$$

$$\leq \left[(\delta_Y + 3)d_X + \delta_X\right](d_Y + 1)(2d_Y^2 - d_Y) - (d_Y^2 - 1)\left[(\delta_Y + 5)d_X + \delta_X\right] + d_Y.$$

Se deduce entonces que

$$\#\{\ x \in I\ /\ f(x) = 0\ \} \leq (d_Y + 1)(2d_Y^2 - d_Y)\big((\delta_Y + 3)d_X + \delta_X\big) + d_Y.$$

□

Observación 2.16 *Notar que en la demostración anterior probamos que cada polinomio ρ_i, con $3 \leq i \leq N + 1$ y τ_i, con $2 \leq i \leq N$, tiene grado acotado por*

$$\deg_X(F_1)d_Y + d_X \deg_Y(F_1) \leq (\delta_Y + 3)d_X d_Y + \delta_X d_Y - d_X.$$

2.2.3. Conteo algorítmico de ceros

En esta sección daremos un algoritmo, que llamaremos `ZeroCounting`, que calcula la cantidad de ceros de una función Pfaffiana $f(x) = F(x, \varphi(x))$ asociada a φ en un intervalo cerrado contenido en el dominio de φ. Para ello supondremos, como es usual en la bibliografía, que disponemos de un oráculo que calcula los signos de funciones Pfaffianas asociadas a φ en números reales algebraicos a partir de su codificación de Thom.

Comenzaremos describiendo una subrutina que nos permitirá calcular los signos de funciones Pfaffianas en un entorno suficientemente chico a la derecha o a la izquierda de un número real algebraico a partir de su codificación de Thom.

Continuando con la misma notación, sean $f(x) = F(x, \varphi(x))$ una función Pfaffiana asociada a φ y $L \in \mathbb{Z}[X]$ no constante. Siguiendo la notación de la Definición 2.11, veamos cómo calcular $\mathrm{sg}(f, \alpha^+)$ y $\mathrm{sg}(f, \alpha^-)$, para toda raíz α de L incluida en el dominio de φ. Notemos, además, $d_X := \deg_X(F)$, $d_Y := \deg_Y(F)$, $\delta_X := \deg_X(\Phi)$ y $\delta_Y := \deg_Y(\Phi)$.

Algoritmo `SignAround`

INPUT: Una función φ que satisface una ecuación diferencial de la forma $\varphi'(x) = \Phi(x, \varphi(x))$, con $\Phi \in \mathbb{Z}[X, Y]$, $\deg_Y(\Phi) > 0$, un polinomio $F \in \mathbb{Z}[X, Y]$ con $\deg_Y(F) > 0$, las codificaciones de Thom de números reales $\alpha_1, \ldots, \alpha_k$ en el dominio de φ como raíces de un polinomio $L \in \mathbb{Z}[X]$, y $\lambda \geq 1$ una cota superior para la multiplicidad de un cero de $f(x) = F(x, \varphi(x))$.
OUTPUT: $\mathrm{sg}(f, \alpha_j^+)$ y $\mathrm{sg}(f, \alpha_j^-)$, $\forall j = 1, \ldots, k$.

1. Calcular los polinomios definidos recursivamente por $F_0 = F$, $F_i = \frac{\partial F_{i-1}}{\partial X} + \frac{\partial F_{i-1}}{\partial Y}\Phi$, para $i = 1, \ldots, \lambda$.

2. Para cada $j = 1, \ldots, k$:

   - Determinar $\nu_j = \min\{0 \leq i \leq \lambda \ / \ g_i(\alpha_j) \neq 0\}$, donde $g_i(x) = F_i(x, \varphi(x))$, utilizando el oráculo.
   - Calcular $\eta_j := \mathrm{sg}(g_{\nu_j}(\alpha_j))$.

   Llamar $\mathrm{sg}(f, \alpha_j^+) = \eta_j$ y $\mathrm{sg}(f, \alpha_j^-) = (-1)^{\nu_j}\eta_j$.

Obtenemos entonces el siguente resultado:

Proposición 2.17 *Sean $f(x) = F(x, \varphi(x))$ una función Pfaffiana asociada a φ y $\alpha_1, \ldots, \alpha_k$ raíces reales de un polinomio $L \in \mathbb{Z}[X]$ que pertenecen al dominio de φ. Sean $d_X = \deg_X(F)$, $d_Y = \deg_Y(F)$, $\delta_X = \deg_X(\Phi)$, $\delta_Y = \deg_Y(\Phi)$ y $\lambda \geq 1$ una cota superior para la multiplicidad de un cero de f. El Algoritmo* `SignAround` *calcula $\mathrm{sg}(f, \alpha_j^+)$ y $\mathrm{sg}(f, \alpha_j^-)$, para todo $j = 1, \ldots, k$, a partir de las codificaciones de Thom de $\alpha_1, \ldots, \alpha_k$ como raíces de L, con una complejidad de orden*

$$O\big(\lambda M(d_X + \lambda\delta_X)M(d_Y + \lambda\delta_Y)\big)$$

y realiza $O(\lambda\, k)$ llamadas al oráculo para calcular el signo en números reales algebraicos de funciones Pfaffianas definidas por polinomios de grado en X acotado por $d_X + \lambda\delta_X$ y grado en Y acotado por $d_Y + \lambda(\delta_Y - 1)$.

Demostración. La correctitud del Algoritmo `SignAround` es consecuencia de que, por la construcción de los polinomios F_i, resulta que $g_i(x) = f^{(i)}$ para $i = 0 \ldots, \lambda$, y de la Observación 2.12.

Analicemos a continuación su complejidad:

1. Por la Proposición 2.3, $\deg_X(F_i) \leq d_X + i\delta_X$ y $\deg_Y(F_i) \leq d_Y + i(\delta_Y - 1)$, para todo $i = 1, \ldots, \lambda$. Luego, calcular cada F_i requiere de:

   - Calcular las derivadas parciales respecto a X y respecto a Y de F_{i-1}. Esto puede hacerse con complejidad $O\Big((d_X + \lambda\delta_X)(d_Y + \lambda\delta_Y)\Big)$.
   - Calcular el producto $\frac{\partial F_{i-1}}{\partial Y}\Phi$. Esto puede hacerse con complejidad $O\Big(M(d_X + \lambda\delta_X)M(d_Y + \lambda\delta_Y)\Big)$.
   - Calcular la suma $\frac{\partial F_{i-1}}{\partial X} + \frac{\partial F_{i-1}}{\partial Y}\Phi$. Esto puede hacerse con complejidad $O\Big((d_X + \lambda\delta_X)(d_Y + \lambda\delta_Y)\Big)$.

   En total, calcular todos los polinomios F_i, para $i = 1, \ldots, \lambda$ requiere de $O\Big(\lambda M(d_X + \lambda\delta_X)M(d_Y + \lambda\delta_Y)\Big)$ operaciones.

2. Para cada $j = 1, \ldots, k$, la determinación del signo en α_j de las funciones g_i, para $i = 0, \ldots, \lambda$, requiere de $O(\lambda)$ llamadas al oráculo para funciones Pfaffianas definidas por polinomios de grado en X acotado por $d_X + \lambda\delta_X$ y grado en Y acotado por $d_Y + \lambda(\delta_Y - 1)$, lo que da un total de $O(\lambda\, k)$ llamadas al oráculo.

□

A continuación describimos el algoritmo más importante de esta sección, el Algoritmo `ZeroCounting`, que nos permite contar la cantidad de ceros de una función Pfaffiana $f(x) = F(x, \varphi(x))$ asociada a φ, con $F \in \mathbb{Z}[X, Y]$, en un intervalo cerrado y acotado incluido en el dominio de φ.

Algoritmo `ZeroCounting`

INPUT: Una función φ que satisface una ecuación diferencial de la forma $\varphi'(x) = \Phi(x, \varphi(x))$, con $\Phi \in \mathbb{Z}[X, Y]$, $\deg_Y(\Phi) > 0$, un polinomio $F \in \mathbb{Z}[X, Y]$ tal que $\mathrm{Res}_Y(F, \widetilde{F}) \neq 0$ y un intervalo cerrado $[a, b] \subset \mathrm{Dom}(\varphi)$ con $a, b \in \mathbb{Q}$.

OUTPUT: La cantidad de ceros de f en $[a, b]$.

1. Calcular el polinomio $F_1(X, Y) := \mathrm{Resto}(\mathrm{cp}(F)^D \widetilde{F}, F)$, donde D es el menor entero par mayor o igual que $1 + \deg_Y(\widetilde{F}) - \deg_Y(F)$ y la división es respecto a la variable Y.

2. Computar los polinomios R_i y τ_i, para $0 \leq i \leq N$, y ρ_i, para $2 \leq i \leq N + 1$, asociados a F y F_1 como en la Notación 2.8.

3. Determinar la lista ordenada de las codificaciones de Thom de las raíces $\alpha_1 < \alpha_2 < \cdots < \alpha_k$ en el intervalo (a, b) de los polinomios τ_i, para $0 \leq i \leq N$, y ρ_i, para $2 \leq i \leq N + 1$.

4. Para todo $0 \leq j \leq k$, siguiendo la Definición 2.9, determinar los signos $\sigma_{I_j, i}$, para $0 \leq i \leq N$, donde $I_j := (\alpha_j, \alpha_{j+1})$, $\alpha_0 = a$ y $\alpha_{k+1} = b$, y computar la secuencia $\mathbf{f}_{I_j} = (f_{I_j, i})_{0 \leq i \leq N}$.

5. Para $0 \leq j \leq k + 1$, decidir si $f(\alpha_j) = 0$ y calcular $\#\{\ 0 \leq j \leq k + 1 : f(\alpha_j) = 0\ \}$.

6. Para todo $0 \leq j \leq k$, computar $v_j := v(\mathbf{f}_{I_j}, \alpha_j^+) - v(\mathbf{f}_{I_j}, \alpha_{j+1}^-)$.

7. Calcular $\#\{0 \leq j \leq k + 1 : f(\alpha_j) = 0\} + \sum_{j=1}^{k} v_j$.

Como consecuencia, obtenemos el siguiente resultado:

Teorema 2.18 *Sea φ una función analítica definida en un abierto $\mathcal{U} \subset \mathbb{R}$ para la cual existe un polinomio $\Phi \in \mathbb{Z}[X, Y]$, con $\deg_Y(\Phi) > 0$, tal que φ satisface en $\mathcal{U}$ la ecuación $\varphi'(x) = \Phi(x, \varphi(x))$. Sea $f(x) = F(x, \varphi(x))$ una función Pfaffiana asociada a φ, con $F \in \mathbb{Z}[X, Y]$. Notemos $d_X = \deg_X(F)$, $d_Y = \deg_Y(F)$, $\delta_X = \deg_X(\Phi)$, $\delta_Y = \deg_Y(\Phi)$ y supongamos que $\mathrm{Res}_Y(F, \widetilde{F}) \neq 0$, donde $\widetilde{F}$ es el polinomio definido en (2.1). El Algoritmo* `ZeroCounting`

calcula la cantidad de ceros de f en un intervalo cerrado $[a,b] \subset Dom(\varphi)$ $(a,b \in \mathbb{Q})$ con complejidad

$$O\Big(d_Y^4 \delta_X \delta_Y (d_Y+\delta_Y)^3 (\delta_Y d_X + \delta_X)^3 \log^3((d_Y+\delta_Y)(\delta_Y d_X+\delta_X))\Big)$$

y requiere de $O\big(d_Y^4(d_Y+\delta_Y)(\delta_Y d_X+\delta_X)^2\big)$ llamadas al oráculo para determinar el signo en números algebraicos reales de funciones Pfaffianas asociadas a φ definidas por polinomios de grado en X de orden $O\big(\delta_X d_Y (d_Y+\delta_Y)(\delta_Y d_X+\delta_X)\big)$ y grado en Y de orden $O\big(\delta_Y d_Y (d_Y+\delta_Y)(\delta_Y d_X+\delta_X\big)$.

Demostración. La correctitud del Algoritmo `ZeroCounting` se sigue de la Proposición 2.10 y del Teorema 2.14.

Para el análisis de la complejidad del Algoritmo `ZeroCounting`:

1. Notar primero que $\deg_Y(F_1) < d_Y$ y que, como $\deg_Y(\widetilde{F}) \geq d_Y$, $D \geq 2$. En la demostración del Corolario 2.15 se probó que $\deg_X(F_1) \leq (\delta_Y+2)d_X+\delta_X$ y, siguiendo con la notación de este corolario, en el paso $i \geq 1$ de la división se obtiene un polinomio $r_i \in \mathbb{Z}[X,Y]$ con $r_i = \mathrm{cp}(F)^{D-i} L_i(X,Y)$, donde $L_i \in \mathbb{Z}[X][Y]$, $\deg_X(L_i) \leq (i+1)d_X+\delta_X$ y

$$L_{i+1}(X,Y) := \mathrm{cp}(F)(X)L_i(X,Y) - \mathrm{cp}(L_i)(X)F(X,Y)Y^{\deg_Y(L_i)-d_Y}. \tag{2.7}$$

   Observar que $\deg_Y(L_i) \leq d_Y+\delta_Y$ para $j = 1,\ldots,i_0 \leq \delta_Y$.

   Por la recurrencia de (2.7), para calcular cada polinomio $L_i \in \mathbb{Z}[X][Y]$, con $i = 1,\ldots,i_0 \leq \delta_Y$, se deben realizar:

   - a lo sumo $\deg_Y(L_i) \leq d_Y+\delta_Y$ productos entre $\mathrm{cp}(F)$ y polinomios en $\mathbb{Z}[X]$ de grado acotado por $(\delta_Y+1)d_X+\delta_X$. Esto puede hacerse con $O\big((d_Y+\delta_Y)M(\delta_Y d_X+\delta_X)\big)$ operaciones.
   - a lo sumo d_Y productos entre los coeficientes de F (como polinomio de variable Y) y el polinomio $\mathrm{cp}(L_i) \in \mathbb{Z}[X]$, de grado acotado por $\deg_X(L_i) \leq (\delta_Y+1)d_X+\delta_X$. Esto puede hacerse con $O\big(d_Y M(\delta_Y d_X+\delta_X)\big)$ operaciones.
   - a lo sumo d_Y restas de polinomios de grado en X acotados por

$$\text{máx}\Big\{\deg_X(\mathrm{cp}(F)L_i), \deg_X(\mathrm{cp}(L_i)FY^{\deg_Y(L_i)-d_Y})\Big\} \leq (\delta_Y+2)d_X+\delta_X.$$

   Luego, la complejidad del i-ésimo paso de la recurrencia es $O\big((d_Y+\delta_Y)M(\delta_Y d_X+\delta_X)\big)$.

   Como multiplicar L_{i_0} por $\mathrm{cp}(F)^{D-i_0}$ no cambia el orden de la complejidad del cálculo de L_{i_0}, resulta que F_1 puede calcularse con complejidad de orden $O\big(\delta_Y(\delta_Y+d_Y)M(\delta_Y d_X+\delta_X)\big)$.

2. Observando las cotas para el grado en X y el grado en Y de F_1 que probamos en el paso 1 y aplicando el algoritmo del Teorema 1.42 a F y a F_1, con $\mu := d_Y d_X(\delta_Y+3)+d_Y\delta_X$, se calculan todas las subresultantes con complejidad de orden

$$O(\mu(d_X d_Y + d_Y(d_X\delta_Y+\delta_X) + d_Y(2d_Y)^{\omega+1}) + d_Y^2 M(\mu)\log(\mu)) \leq$$

$$\leq O\Big((d_Y(\delta_Y d_X+\delta_X)(d_Y(\delta_Y d_X+\delta_X)+d_Y^{\omega+2})+d_Y^2\mu\Big).$$

Observando que

$$O(d_Y^2(\delta_Y d_X+\delta_X)^2+d_Y^{\omega+3}(\delta_Y d_X+\delta_X))\leq O\Big(d_Y^{\omega+3}(\delta_Y d_X+\delta_X)^2\Big),$$

obtenemos una cota para la complejidad de este paso.

3. Consideremos el polinomio L definido por

$$L(X):=\prod_{0\leq i\leq N}\tau_i\prod_{3\leq i\leq N+1}\rho_i. \tag{2.8}$$

En la demostración del Corolario 2.15 se hallaron cotas superiores de N y de los grados de los polinomios τ_i, para $i=0,\ldots,N$, y ρ_i, para $i=3,\ldots,N+1$ (ver Observación 2.16), de donde se deduce que

$$\deg(L)\leq(2d_Y^2-d_Y)((\delta_Y+3)d_X+\delta_X).$$

Calculamos sus coeficientes por interpolación: la especialización de L en un punto puede ser calculada con $O(d_Y^2(\delta_Y d_X+\delta_X))$ operaciones de especializar sus factores y multiplicarlos. La complejidad de interpolar en $\deg(L)+1$ puntos es de orden $O(d_Y^4(\delta_Y d_X+\delta_X)^2)$.

Aplicar el algoritmo del Teorema 1.26 a L, las derivadas sucesivas de L, $X-a$ y $b-X$ para calcular y ordenar las codificaciones de Thom de las raíces de L en (a,b) tiene complejidad

$$O\big(d_Y^6(\delta_Y d_X+\delta_X)^3\log^3(d_Y^2(\delta_Y d_X+\delta_X))\big).$$

La complejidad total de este paso es de orden $O\Big(d_Y^6(\delta_Y d_X+\delta_X)^3\log^3(d_Y(\delta_Y d_X+\delta_X))\Big)$.

4. Para calcular los signos de los polinomios $(\rho_i)_{i=2,\ldots,N+1}$ y $(\tau_i)_{i=0,\ldots,N}$ entre dos raíces consecutivas de L, según la Observación 2.12, necesitamos calcular los signos de $O\big(Nd_Y((\delta_Y+3)d_X+\delta_X)\big)$ polinomios (los anteriores y sus derivadas sucesivas) en cada una de las raíces de L en (a,b). Recordando que $N\leq d_Y$, por el Teorema 1.22, puede hacerse con complejidad de orden

$$O(d_Y^6(\delta_Y d_X+\delta_X)^3\log^3(d_Y^6(\delta_Y d_X+\delta_X)^3))=O(d_Y^6(\delta_Y d_X+\delta_X)^3\log^3(d_Y(\delta_Y d_X+\delta_X))).$$

A partir de ellos, la secuencia $\mathbf{f}_{I_j}$ puede calcularse sin cambio en la complejidad.

La complejidad total de los pasos 1 a 4 es de orden

$$O\Big(d_Y^6(\delta_Y d_X+\delta_X)^3\log^3(d_Y(\delta_Y d_X+\delta_X))\Big).$$

5. y 6. Estos pasos requieren determinar el signo de funciones Pfaffianas de la forma $G(x,\varphi(x))$, con $G\in\mathbb{Z}[X,Y]$, en números algebraicos dados por su codificación de Thom como raíz de L y en a y b.

En el paso 5, necesitamos $k+2 \leq \deg(L)+2 = O(d_Y^2(\delta_Y d_X + \delta_X))$ llamadas al oráculo para la función Pfaffiana definida por el polinomio F, con $\deg_X(F) = d_X$ y $\deg_Y(F) = d_Y$.

En el paso 6, se aplica el Algoritmo `SignAround` (ver Proposición 2.17) a las funciones f_i, para cada $i = 0, \ldots, N$ utilizando el polinomio L definido en el paso 3. Dado que para todo $i = 0, \ldots, N$, por la Proposición 1.38, $\deg_X(R_i) \leq d_Y\big((\delta_Y+3)d_X + \delta_X\big) - d_X$ y $\deg_Y(R_i) \leq d_Y$, usando la Proposición 2.4, se tiene que

$$\begin{aligned}
&\operatorname{mult}(\alpha_j, f_i) \leq 2\deg_X(R_i)\deg_Y(R_i) + \deg_X(R_i)(\delta_Y - 1) + \deg_Y(R_i)(\delta_X + 1) \leq \\
&\leq (2d_Y^2 + d_Y(\delta_Y - 1))((\delta_Y+3)d_X + \delta_X) - 2d_X d_Y - d_X\delta_Y + d_X + d_Y(\delta_X+1) \leq \\
&\leq d_Y(2d_Y + \delta_Y - 1)((\delta_Y+3)d_X + \delta_X) - 2d_X d_Y - d_X(\delta_Y - 1) + d_Y(\delta_X+1) \leq \\
&\leq d_Y(2d_Y + \delta_Y)((\delta_Y+3)d_X + \delta_X) + d_Y := \lambda,
\end{aligned}$$

para todo $j = 0, \ldots, \deg(L)$, para todo $i = 0, \ldots, N$.

Como para todo $i = 0, \ldots, N$:

$$\begin{cases}
\deg_X(R_i) + \lambda\delta_X = O\big(\delta_X d_Y(d_Y+\delta_Y)(\delta_Y d_X + \delta_X)\big) \\
\deg_Y(R_i) + \lambda\delta_Y = O\big(\delta_Y d_Y(d_Y+\delta_Y)(\delta_Y d_X + \delta_X)\big) \\
\lambda\deg(L) = O\big(d_Y^3(d_Y+\delta_Y)(d_X\delta_Y + \delta_X)^2\big)
\end{cases}$$

se tiene que, aplicar $N \leq d_Y$ veces el algoritmo `SignAround` requiere del orden de $O(d_Y\lambda M(\deg_X(R_i)+\lambda\delta_X)M(\deg_Y(R_i)+\lambda\delta_Y)) \leq O\Big(\delta_X\delta_Y d_Y^4(\delta_Y+d_Y)^3\ (\delta_Y d_X+\delta_X)^3 \times \log^2((d_Y+\delta_Y)(\delta_Y d_X+\delta_X))\log^2(\log((d_Y+\delta_Y)(\delta_Y d_X+\delta_X))\Big)$, lo que resulta en una complejidad de orden $O\Big(\delta_X\delta_Y d_Y^4(\delta_Y+d_Y)^3\ (\delta_Y d_X+\delta_X)^3\log^3((d_Y+\delta_Y)(\delta_Y d_X+\delta_X))\Big)$, con $O(d_Y^4(d_Y+\delta_Y)(d_X\delta_Y+\delta_X)^2)$ llamadas al oráculo para calcular el signo en números reales algebraicos de funciones Pfaffianas definidas por polinomios de grado en X de orden $O(\delta_X d_Y(d_Y+\delta_Y)(\delta_Y d_X+\delta_X))$ y grado en Y de orden $O(\delta_Y d_Y(d_Y+\delta_Y)(\delta_Y d_X + \delta_X))$.

La complejidad total del algoritmo es de orden

$$O\Big(d_Y^4\delta_X\delta_Y(d_Y+\delta_Y)^3(\delta_Y d_X+\delta_X)^3\log^3((d_Y+\delta_Y)(\delta_Y d_X+\delta_X))\Big),$$

y se efectúan $O(d_Y^4(d_Y+\delta_Y)(\delta_Y d_X+\delta_X)^2)$ llamadas al oráculo para calcular el signo en números reales algebraicos de funciones Pfaffianas definidas por polinomios de grado en X de orden de $O(\delta_X d_Y(d_Y+\delta_Y)(\delta_Y d_X+\delta_X))$ y grado en Y de orden de $O(\delta_Y d_Y(d_Y+\delta_Y)(\delta_Y d_X + \delta_X))$. □

2.3. Secuencia de Sturm generalizada

En esta sección, introduciremos la noción de secuencia de Sturm de una función continua respecto a otra. Veremos cómo esta secuencia se relaciona con los indicadores de Tarski de las funciones involucradas generalizando el Teorema 1.6 (válido para polinomios) y cómo, bajo ciertas hipótesis, esta secuencia puede ser calculada algorítmicamente.

2.3.1. Definición y propiedad fundamental

Comenzaremos proponiendo una definición para secuencia de Sturm de una función respecto a otra en un intervalo abierto y probaremos la propiedad fundamental que verifica.

Definición 2.19 *Sean $f, g : (a, b) \to \mathbb{R}$ funciones continuas. Una* secuencia de Sturm en (a, b) para f respecto a g *es una secuencia finita de funciones continuas $\mathbf{f} = (f_0, f_1, \ldots, f_N)$ definidas en (a, b) que cumple que para todo $y \in (a, b)$:*

1) $f_0(y) = 0 \iff f(y) = 0 \ \wedge \ g(y) \neq 0$.

2) *Si $f_0(y) = 0$ entonces $\exists\ \varepsilon > 0$ tal que $f_1(x) \neq 0\ \forall x \in (y - \varepsilon, y + \varepsilon)$.*

   En este intervalo, si $g(y) > 0$ se tiene que

$$\begin{cases} f_0(x)f_1(x) < 0 & para \ \ y - \varepsilon < x < y \\ f_0(x)f_1(x) > 0 & para \ \ y < x < y + \varepsilon \end{cases}$$

   y si $g(y) < 0$ se tiene que

$$\begin{cases} f_0(x)f_1(x) > 0 & para \ \ y - \varepsilon < x < y \\ f_0(x)f_1(x) < 0 & para \ \ y < x < y + \varepsilon. \end{cases}$$

3) *Para todo $i = 1, \ldots, N - 1$, si $f_i(y) = 0$ entonces $f_{i-1}(y)f_{i+1}(y) < 0$.*

4) $f_N(y) \neq 0$.

Teorema 2.20 *Si $\mathbf{f} = (f_0, \ldots, f_N)$ es una secuencia de Sturm para f respecto a g en (a, b) y $c, d \in (a, b)$ no son ceros de f, si $c < d$, entonces*

$$v(\mathbf{f}, c) - v(\mathbf{f}, d) = \#\left\{x \in (c, d) / \ f(x) = 0 \wedge g(x) > 0\right\} - \#\left\{x \in (c, d) / \ f(x) = 0 \wedge g(x) < 0\right\}.$$

Demostración. Sea $t \in [c, d] \subset (a, b)$. Por la continuidad de las funciones $f_0, \ldots, f_N$ y por la condición 3 de la Definición 2.19, existe $\varepsilon > 0$ tal que si $a \leq t - \varepsilon < x < t \leq y < t + \varepsilon \leq b$:

$$\forall i \neq 0, N \text{ tal que } f_i(t) = 0 \text{ vale } \begin{cases} \operatorname{sg}(f_{i-1}(x)) = \operatorname{sg}(f_{i-1}(y)) \neq 0 \\ \operatorname{sg}(f_{i+1}(x)) = \operatorname{sg}(f_{i+1}(y)) \neq 0 \\ \operatorname{sg}(f_{i-1}(x)) = -\operatorname{sg}(f_{i+1}(x)) \neq 0 \\ \operatorname{sg}(f_{i-1}(y)) = -\operatorname{sg}(f_{i+1}(y)) \neq 0 \end{cases} \tag{2.9}$$

y

$$\forall j \text{ tal que } f_j(t) \neq 0 \text{ vale } \ \operatorname{sg}(f_j(x)) = \operatorname{sg}(f_j(t)) = \operatorname{sg}(f_j(y)). \tag{2.10}$$

Notar que, por la condición 4 de la Definición 2.19, se verifica (2.10) para $j = N$.

Para simplificar la notación, notemos $v(x) = v(\mathbf{f}, x)$ y analicemos la función $v(x)$. Consideremos las siguientes opciones:

- Si $f_0(t) \neq 0$, por (2.9) y (2.10), se tiene que $v(x) = v(y)\ \forall x, y \in (t-\varepsilon, t+\varepsilon)$. Luego,

$$v(x) = v(t)\ \forall x \in (t-\varepsilon, t+\varepsilon). \tag{2.11}$$

- Si $f_0(t) = 0$, por la condición 2 de la Definición 2.19, $f_1(t) \neq 0$ y, para ε suficientemente chico, no cambia de signo en $(t-\varepsilon, t+\varepsilon)$. Además:

  si $g(t) > 0$ se tiene que

$$\begin{cases} f_0(x)f_1(x) < 0 & \text{para } t-\varepsilon < x < t \\ f_0(x)f_1(x) > 0 & \text{para } t < x < t+\varepsilon, \end{cases}$$

  y si $g(t) < 0$ se tiene que

$$\begin{cases} f_0(x)f_1(x) > 0 & \text{para } t-\varepsilon < x < t \\ f_0(x)f_1(x) < 0 & \text{para } t < x < t+\varepsilon. \end{cases}$$

  Se obtiene entonces que si $a \leq t-\varepsilon < x < t < y < t+\varepsilon \leq b$, además de (2.9) y (2.10):

$$\left\{ \begin{array}{l} \text{sg}(f_1(x)) = \text{sg}(f_1(y)) \neq 0,\ y \\ \text{si}\ \ g(t) > 0 \text{ vale que } \left\{ \begin{array}{l} \text{sg}(f_0(x)) = -\text{sg}(f_1(x)) \neq 0 \\ \text{sg}(f_0(y)) = \text{sg}(f_1(y)) \neq 0 \end{array} \right. \quad y \\ \text{si}\ \ g(t) < 0 \text{ vale que } \left\{ \begin{array}{l} \text{sg}(f_0(x)) = \text{sg}(f_1(x)) \neq 0 \\ \text{sg}(f_0(y)) = -\text{sg}(f_1(y)) \neq 0. \end{array} \right. \end{array} \right.$$

  Luego, si $g(t) > 0$, $v(x) = v(y) + 1$, y si $g(t) < 0$, $v(x) = v(y) - 1$.

  Además, como $f_0(t) = 0$, se tiene que

$$v(t) = \begin{cases} v(x) - 1 & \text{si } g(t) > 0 \\ v(x) & \text{si } g(t) < 0 \end{cases} \quad \forall x \in (t-\varepsilon, t).$$

  Se obtiene entonces que

$$\begin{array}{ll} \text{Si } g(t) > 0: & v(x) = \begin{cases} v(t)+1 & \forall x \in (t-\varepsilon, t) \\ v(t) & \forall x \in [t, t+\varepsilon) \end{cases} \\ \text{Si } g(t) < 0: & v(x) = \begin{cases} v(t) & \forall x \in (t-\varepsilon, t] \\ v(t)+1 & \forall x \in (t, t+\varepsilon). \end{cases} \end{array} \tag{2.12}$$

Se construye así un cubrimiento de $[c,d]$ por intervalos abiertos $(I_t)_{t\in[c,d]}$ centrados en t, para los cuales o bien $f_0(t) \neq 0$ y se verifica (2.11), o bien $f_0(t) = 0$ y se verifica (2.12). Como $[c,d]$ es compacto, existen $c \leq t_1 < t_2 < \ldots < t_k \leq d$ tales que $[c,d] = \bigcup_{i=1,\ldots,k} I_{t_i}$. Podemos suponer que ninguno de estos intervalos está contenido en otro, y por lo tanto, $I_{t_i} \cap I_{t_{i+1}} \neq \emptyset, \forall i = 1, \ldots, k-1$.

Renombrando $r_1 < \ldots < r_s$ a los valores de t_i que son ceros de f_0, y por lo tanto no de g, se obtienen las siguientes observaciones:

1. $c < r_1$ y $r_s < d$, pues c y d no son ceros de f y por lo tanto, tampoco de f_0.

2. Si $t \in [c,d]$, $t \neq r_j \ \forall j = 1,\dots,s$ entonces $f_0(t) \neq 0$: existe t_i, con $i = 1,\dots,k$ tal que $t \in I_{t_i}$, pero por construcción de I_{t_i}, f_0 no se anula allí salvo quizás en t_i.

3. $\{x \in (c,d) \ / \ f(x) = 0,\ g(x) > 0\} = \{x \in (c,d) \ / \ f_0(x) = 0,\ g(x) > 0\}$ que por la observación anterior es igual a $\{r_j / \ g(r_j) > 0\}$.
   Análogamente, $\{x \in (c,d) \ / \ f(x) = 0,\ g(x) < 0\} = \{r_j / \ g(r_j) < 0\}$.

4. v es constante en (r_j, r_{j+1}), $\forall j = 1,\dots,s$, pues $(I_{t_i})_{i=1,\dots,k}$ es un cubrimiento de (c,d) con intervalos que verifican (2.11) y (2.12).

Para cada $j = 1,\dots,s-1$, tomemos $\xi_j \in (r_j, r_{j+1})$, y sean $\xi_0 = c$ y $\xi_s = d$. Observemos que $v(c) = v(\xi_0)$ y $v(\xi_s) = v(d)$, pues $c \in I_{r_1}$, $d \in I_{r_s}$ y v es constante en la primera mitad y en la segunda mitad de estos intervalos. Además, dado que se verifican (2.11) y (2.12), se tiene que $\forall\ j = 1,\dots,s$,

$$v(\xi_{j-1}) - v(\xi_j) = \begin{cases} 1 & \text{si } g(r_j) > 0 \\ -1 & \text{si } g(r_j) < 0. \end{cases}$$

Luego,

$$v(c) - v(d) = \sum_{j=1}^{s} v(\xi_{j-1}) - v(\xi_j) = \#\{r_j / \ g(r_j) > 0\} - \#\{r_j / \ g(r_j) < 0\}.$$

□

De la misma forma que lo hicimos en la Sección 2.2, comenzaremos a trabajar en particular con funciones Pfaffianas asociadas a una función φ. Para ello, supondremos que φ es una función fija definida en un dominio $\mathcal{U} \subset \mathbb{R}$ que satisface en $\mathcal{U}$ una ecuación diferencial de la forma $\varphi'(x) = \Phi(x, \varphi(x))$, donde $\Phi \in \mathbb{Z}[X,Y]$ con $\deg_Y(\Phi) > 0$.

A continuación, generalizamos la definición de indicador de Tarski para polinomios (ver Definición 1.5) a funciones Pfaffianas asociadas a φ:

Definición 2.21 *Sean $f(x) = F(x, \varphi(x))$ y $g(x) = G(x, \varphi(x))$ funciones Pfaffianas asociadas a φ definidas por polinomios $F, G \in \mathbb{Z}[X,Y]$, con $\deg_Y(F) > 0$. Sea $(c,d) \subset \mathbb{R}$ un intervalo abierto y acotado tal que $[c,d]$ está contenido en el dominio de φ. Definimos el* indicador de Tarski de f para g en (c,d) *como el número*

$$\mathrm{TaQ}(f,g;c,d) := \#\{x \in (c,d) \ / \ f(x) = 0 \wedge g(x) > 0\} - \#\{x \in (c,d) \ / \ f(x) = 0 \wedge g(x) < 0\}.$$

Con esta nueva notación, el Teorema 2.20 aplicado a funciones Pfaffianas asociadas a φ se reescribe de la siguiente forma:

Teorema 2.22 *Sean f y g funciones Pfaffianas asociadas a φ definidas en un intervalo abierto y acotado $(a,b) \subset \mathrm{Dom}(\varphi)$. Si $\mathbf{f} = (f_0,\dots,f_N)$ es una secuencia de Sturm para f respecto a g en (a,b) y $c,d \in (a,b)$ no son ceros de f, si $c < d$, entonces*

$$v(\mathbf{f}, c) - v(\mathbf{f}, d) = \mathrm{TaQ}(f,g;c,d).$$

2.3.2. Construcción de un secuencia de Sturm generalizada

En esta sección veremos cómo, dadas dos funciones Pfaffianas f y g asociadas a φ y definidas por polinomios $F, G \in \mathbb{Z}[X, Y]$, podemos construir una secuencia de Sturm para f respecto a g en un intervalo acotado I contenido en el dominio de φ, bajo ciertas hipótesis sobre I. A continuación, aplicaremos esta construcción para calcular indicadores de Tarski.

Sean $f(x) = F(x, \varphi(x))$ y $g(x) = G(x, \varphi(x))$ funciones Pfaffianas asociadas a φ, donde $F, G \in \mathbb{Z}[X, Y]$, con $\deg_Y(F), \deg_Y(G) > 0$.

Consideremos el polinomio $\widetilde{F} \in \mathbb{Z}[X, Y]$ definido en (2.1),

$$\widetilde{F}(X, Y) = \frac{\partial F}{\partial X}(X, Y) + \frac{\partial F}{\partial Y}(X, Y)\Phi(X, Y).$$

Recordemos que

$$f'(x) = \widetilde{F}(x, \varphi(x)).$$

Dado que el polinomio $\widetilde{F}G \in \mathbb{Z}[X, Y]$ tendrá un papel importante en el algoritmo que desarrollaremos, será útil tener en consideración las siguientes desigualdades respecto de su grado en X y en Y:

Lema 2.23 *Con las hipótesis y notaciones anteriores, si* $\delta_Y = \deg_Y(\Phi)$ *y* $\delta_X = \deg_X(\Phi)$

1) $\deg_Y(F) < \deg_Y(\widetilde{F}G) \leq \deg_Y(G) + \deg_Y(F) + \delta_Y - 1$

2) $\deg_X(\widetilde{F}G) \leq \deg_X(G) + \deg_X(F) + \delta_X$.

Demostración. Ambas desigualdades se siguen trivialmente del hecho de que $\deg_Y(F) \leq \deg_Y(\widetilde{F}) \leq \deg_Y(F) + \delta_Y - 1$ (ver las desigualdades en (2.2)) y de que $\deg_Y(G) > 0$. □

Podemos entonces, aplicar la construcción del Corolario 1.40 a los polinomios $\widetilde{F}G$ y F, utilizando por conveniencia una notación con los índices corridos respecto a los del corolario antes mencionado:

Notación 2.24 *Con la notación anterior se definen:*

- $n_{-1} := \deg_Y(\widetilde{F}G) + 1, \quad R_{-1} := \widetilde{F}G$;
- $n_0 := \deg_Y(\widetilde{F}G), \quad R_0 := F$;
- *para* $i \geq 0$, *si* $R_i \neq 0$,

$$n_{i+1} := \deg_Y(R_i) \text{ y } R_{i+1} := \mathrm{SRes}_{n_{i+1}-1}(\widetilde{F}G, F) \in \mathbb{Z}[X][Y].$$

Sea $N := \max\{i \geq 0 \mid R_i \neq 0\}$. *Notamos, para* $i = -1, \ldots, N$, $\tau_i \in \mathbb{Z}[X]$ *el coeficiente principal de* R_i *y, para* $i = 1, \ldots, N+1$, $\rho_i = \mathrm{sRes}_{n_i}(\widetilde{F}G, F) \in \mathbb{Z}[X]$.

Como consecuencia del Corolario 1.40, resulta que $N \leq \deg_Y(F)$ y se obtienen también las siguientes relaciones:

$$\beta\ \tau_0^{\deg_Y(\widetilde{F}G)-\deg_Y(F)+1} R_{-1} = C_0 R_0 - R_1, \tag{2.13}$$

$$\rho_{i+2}\tau_{i+1}R_i = C_{i+1}R_{i+1} - \rho_{i+1}\tau_i R_{i+2},\ 0 \leq i \leq N-1, \tag{2.14}$$

donde $\beta = (-1)^{\frac{1}{2}(\deg_Y(\widetilde{F}G)-\deg_Y(F))(\deg_Y(\widetilde{F}G)-\deg_Y(F)-1)}$.

De aquí se deduce que el polinomio $R_N(X,Y)$ divide a los polinomios $R_i(X,Y)$ en $\mathbb{Q}(X)[Y]$ para todo $i = -1,\ldots,N$. Luego, $P_i = \dfrac{R_i}{R_N} \in \mathbb{Q}(X)[Y]$, para $i = -1,\ldots,N$.

Definición 2.25 *Dado un intervalo abierto I contenido en el dominio de φ y que no contenga raíces de τ_i para $i = 0,\ldots,N$ ni de ρ_i para $i = 1,\ldots,N+1$, se define inductivamente una secuencia de signos $(\sigma_{I,i})_{0\leq i\leq N} \in \{1,-1\}^{N+1}$ de la siguiente manera:*

- $\sigma_{I,0} = 1$,
- $\sigma_{I,1} = -\beta \mathrm{sg}_I(\tau_0)^{\deg_Y(\widetilde{F}G)-\deg_Y(F)+1}$
- $\sigma_{I,i+2} = \mathrm{sg}_I(\rho_{i+2}\tau_{i+1}\rho_{i+1}\tau_i)\sigma_{I,i}$, *si* $0 \leq i \leq N-2$.

Se introducen además, las funciones pfaffianas

$$f_{I,i}(x) = \sigma_{I,i} P_i(x,\varphi(x)),\ \textit{para}\ 0 \leq i \leq N.$$

Teorema 2.26 *Con las hipótesis y la notación de la Definición 2.25, la secuencia de funciones Pfaffianas $\mathbf{f}_I = (f_{I,i})_{0\leq i\leq N}$ resulta ser una secuencia de Sturm para f respecto a g en el intervalo I.*

Demostración. Para simplificar la notación suprimiremos el sufijo I en la escritura de $f_{I,i}$ y de $\sigma_{I,i}$ y llamaremos $r(x) = R_N(x,\varphi(x))$. Observemos primero que $\mathbf{f}_I$ es una secuencia de funciones continuas pues $P_i = \dfrac{R_i}{R_N} \in \mathbb{Q}(X)[Y]$ y los únicos denominadores que aparecen son potencias de τ_N, que no se anulan en I por hipótesis. Observemos además que

$$f(x) = f_0(x)r(x) \text{ y } f'(x)g(x) = f_{-1}(x)r(x). \tag{2.15}$$

Veamos que se cumplen las cuatro condiciones de la Definicion 2.19.

1) Veamos que $f_0(y) = 0 \iff f(y) = 0$ y $g(y) \neq 0$:

   $\Rightarrow$) Si $f_0(y) = 0$, claramente $f(y) = 0$ y $\mathrm{mult}(y,r) < \mathrm{mult}(y,f)$.
   Por otro lado, teniendo en cuenta la Observación 1.35, existen polinomios $A, B \in \mathbb{Z}[X,Y]$ tales que $R_N = A\widetilde{F}G + BF$. Evaluando en $(x,\varphi(x))$ obtenemos que

   $$r(x) = A(x,\varphi(x))f'(x)g(x) + B(x,\varphi(x))f(x). \tag{2.16}$$

   Si $g(y) = 0$, se obtiene que $\mathrm{mult}(y,f) \leq \mathrm{mult}(y,r)$. Obtenemos así la contradicción

   $$\mathrm{mult}(y,r) < \mathrm{mult}(y,f) \leq \mathrm{mult}(y,r).$$

   Luego, $g(y) \neq 0$.

⇐) Si $f(y) = 0$ y $g_0(y) \neq 0$, entonces $\text{mult}(y, f'g) = \text{mult}(y, f) - 1$. Además, por (2.15), se tiene que

$$\text{mult}(y, r) \leq \text{mult}(y, f'g) = \text{mult}(y, f) - 1 = \text{mult}(y, f_0) + \text{mult}(y, r) - 1,$$

de lo que se deduce que $1 \leq \text{mult}(y, f_0)$. Luego, $f_0(y) = 0$.

2) Como $f_0(y) = 0$, en virtud de (2.15), $f(y) = 0$. Como es una función analítica, existen $m \in \mathbb{N}$ y una función analítica α que no se anula en y tales que $f(x) = (x-y)^m\alpha(x)$, y por lo tanto, $f'(x) = (x-y)^{m-1}(m\alpha(x) + (x-y)\alpha'(x))$. Llamando $\gamma(x) = m\alpha(x) + (x-y)\alpha'(x)$, obtenemos que $\gamma(y) = m\alpha(y)$. Luego, γ y α tienen el mismo signo en un entorno de y en I.
Veamos que $\text{mult}(y, r) = m - 1$.
Observemos primero que $\text{mult}(y, f'g) = m - 1$ dado que $g(y) \neq 0$ (pues se verifica la condición 1). Por un lado, en virtud de (2.15), $\text{mult}(y, r) \leq m - 1$. Por otro lado, por la identidad (2.16), $\text{mult}(y, r) \geq \min\{\text{mult}(y, f'g), \text{mult}(y, f)\} = m - 1$. Existe entonces una función s analítica tal que $r(x) = (x-y)^{m-1}s(x)$ y $s(y) \neq 0$.
Luego, por (2.15), obtenemos que

$$f_0(x)f_{-1}(x) = \frac{f(x)f'(x)g(x)}{r^2(x)} = \frac{(x-y)^m\alpha(x)(x-y)^{m-1}\gamma(x)g(x)}{\big((x-y)^{m-1}s(x)\big)^2} =$$

$$= \frac{(x-y)\alpha(x)\gamma(x)g(x)}{s^2(x)}.$$

Se deduce que, como $g(y), \alpha(y), \gamma(y), s(y) \neq 0$, $\text{mult}(y, f_0) + \text{mult}(y, f_{-1}) = 1$. Luego, como $f_0(y) = 0$, $f_{-1}(y) \neq 0$. Como $\gamma(y) = m\alpha(y)$, obtenemos entonces que, para x suficientemente cerca de y, $\text{sg}(f_0(x)f_{-1}(x)) = \text{sg}((x-y)g(x))$. Por otro lado, dividiendo por $R_N(X, Y)$, multiplicando por σ_1 y evaluando en $Y = \varphi(x)$ en la recurrencia (2.13), se tiene que

$$\sigma_1\beta\tau_0^{\deg_Y(\widetilde{F}G)-\deg_Y(F)+1}f_{-1}(x) = \sigma_1 C_0(x, \varphi(x))f_0(x) - f_1(x).$$

Evaluando en y, obtenemos que $\sigma_1\beta\tau_0^{\deg_Y(\widetilde{F}G)-\deg_Y(F)+1}f_{-1}(y) = -f_1(y)$. Por la definición de σ_1, se tiene que $\text{sg}(f_{-1}(y)) = \text{sg}(f_1(y))$. Como $f_{-1}(y) \neq 0$, se concluye que en un entorno de y en I:

$$\text{sg}\big(f_0(x)f_1(x)\big) = \text{sg}\big(f_0(x)f_{-1}(x)\big) = \text{sg}\big((x-y)g(x)\big).$$

Luego, si $g(y) > 0$ entonces $\text{sg}(f_0f_1) = \text{sg}(x-y)$ en un entorno de y y si $g(y) < 0$ entonces $\text{sg}(f_0f_1) = -\text{sg}(x-y)$ como se quería probar.

3) Sean $i \in \{1, \ldots, N-1\}$, $y \in I$ tales que $f_i(y) = 0$. Veamos que $f_{i-1}(y)f_{i+1}(y) < 0$.

Observemos que $R_i(y, \varphi(y)) = 0$ y dada la fórmula (2.14), se tiene que

$$R_{i+1}(y, \varphi(y)) = \frac{\rho_{i+1}(y)\tau_i(y)R_{i-1}(y, \varphi(y))}{-\rho_i(y)\tau_{i-1}(y)}.$$

Luego, se obtiene que

$$f_{i-1}(y)f_{i+1}(y) = \frac{\sigma_{i-1}R_{i-1}(y,\varphi(y))}{R_N(y,\varphi(y))}\ \frac{\sigma_{i+1}R_{i+1}(y,\varphi(y))}{R_N(y,\varphi(y))} =$$

$$= \frac{\sigma_{i-1}\sigma_{i+1}\rho_{i+1}(y)\tau_i(y)}{-\rho_i(y)\tau_{i-1}(y)}\Big(\frac{f_{i-1}(y)}{\sigma_{i-1}}\Big)^2.$$

Como $f_{i-1}(y) \neq 0$ (pues sino por la recurrencia (2.14) se tendrá que $f_{-1}(y) = f_0(y) = 0$ y vimos en la demostración de 1) que eso no puede pasar) y las funciones $\rho_{i+1}, \tau_i, \rho_i$ y τ_{i-1} no se anulan en I,

$$\mathrm{sg}\big(f_{i-1}(y)f_{i+1}(y)\big) = -\sigma_{i-1}\sigma_{i+1}\mathrm{sg}\Big(\frac{\rho_{i+1}(y)\tau_i(y)}{\rho_i(y)\tau_{i-1}(y)}\Big) = -\sigma_{i+1}^2 = -1,$$

como se quería probar.

4) $f_N(y) = \sigma_N \neq 0$, $\forall y \in I$.

□

Por el teorema anterior y el Teorema 2.20, se tiene que:

Corolario 2.27 *Si I es un intervalo contenido en el dominio de φ que no contiene raíces de τ_i para $i = 0, \ldots, N$ ni de ρ_i para $i = 1, \ldots, N+1$ y $c, d \in I$, $c < d$, no son ceros de f,*

$$v(\mathbf{f}_I, c) - v(\mathbf{f}_I, d) = \mathrm{TaQ}(f, g; c, d).$$

En vista de generalizar este resultado a intervalos abiertos incluidos en el dominio de φ y de evitar la división por R_N, realizamos la siguiente observación:

Observación 2.28 *Con las mismas hipótesis y notación que en la Definición 2.25, sea $\mathbf{p}_I = (p_{I,i})_{0\leq i\leq N}$, donde $p_{I,i}(x) = \sigma_{I,i}R_i(x,\varphi(x))$ para $i = 0, \ldots, N$. Sea $\xi \in I = (a, b)$ y supongamos que $R_N(\xi, \varphi(\xi)) \neq 0$. Se tiene que $v(\mathbf{p}_I, \xi) = v(\mathbf{f}_I, \xi)$. Luego, teniendo en cuenta la Definición 2.11, $v(\mathbf{p}_I, a^+) = v(\mathbf{f}_I, a^+)$ y $v(\mathbf{p}_I, b^-) = v(\mathbf{f}_I, b^-)$.*

Por el Teorema 2.20, obtenemos:

Proposición 2.29 *Con las mismas hipótesis y notaciones que la Observación 2.28, si además el intervalo cerrado $[a, b]$ está contenido en el dominio de φ, se tiene que*

$$v(\mathbf{p}_I, a^+) - v(\mathbf{p}_I, b^-) = \mathrm{TaQ}(f, g; a, b).$$

Como consecuencia obtenemos una versión generalizada del Teorema 2.26:

Teorema 2.30 *Sea φ una función analítica que satisface una ecuación diferencial de la forma $\varphi'(x) = \Phi(x, \varphi(x))$ para un polinomio $\Phi \in \mathbb{Z}[X, Y]$ con $\deg_Y(\Phi) > 0$. Sea $(a, b) \subset \mathbb{R}$ un intervalo abierto y acotado tal que $[a, b]$ está contenido en el dominio de φ. Sean $f(x) = F(x, \varphi(x))$ y $g(x) = G(x, \varphi(x))$ funciones Pfaffianas asociadas a φ definidas por polinomios $F, G \in \mathbb{Z}[X, Y]$ con $\deg_Y(F) > 0$ y $\deg_Y(G) > 0$.*

Sean ρ_i y τ_i los polinomios en $\mathbb{Z}[X]$ introducidos en la Notación 2.24 y $\alpha_1 < \alpha_2 < \ldots < \alpha_k$ todas sus raíces en (a, b). Entonces el valor de $\mathrm{TaQ}(f, g; a, b)$ *coincide con*

$$\#\{\ 1 \leq j \leq k : f(\alpha_j) = 0 \wedge g(\alpha_j) > 0\ \} - \#\{\ 1 \leq j \leq k : f(\alpha_j) = 0 \wedge g(\alpha_j) < 0\ \} +$$

$$+ \sum_{j=0}^{k} v(\mathbf{p}_{I_j}, \alpha_j^+) - v(\mathbf{p}_{I_j}, \alpha_{j+1}^-),$$

donde $\alpha_0 = a$, $\alpha_{k+1} = b$ y, para todo $0 \leq j \leq k$, $I_j = (\alpha_j, \alpha_{j+1})$ y $\mathbf{p}_{I_j}$ es la secuencia de funciones $p_{I_j,i}(x) = \sigma_{I_j,i} R_i(x, \varphi(x))$ para $i = 0, \ldots, N$ con $\sigma_{I_j,i}$ introducidos en la Definición 2.25 y $R_i(X, Y)$ definidos en la Notación 2.24.

2.3.3. Cálculo efectivo de indicadores de Tarski

Como consecuencia del Teorema 2.30, podemos describir un algoritmo para calcular el número $\mathrm{TaQ}(f, g; a, b)$.

Algoritmo `Tarski-query`

INPUT: Una función φ que satisface una ecuación diferencial $\varphi'(x) = \Phi(x, \varphi(x))$, con $\Phi \in \mathbb{Z}[X, Y]$, $\deg_Y(\Phi) > 0$; polinomios $F, G \in \mathbb{Z}[X, Y]$ con $\deg_Y(F), \deg_Y(G) > 0$, y un intervalo cerrado $[a, b] \subset \mathrm{Dom}(\varphi)$.

OUTPUT: $\mathrm{TaQ}(f, g; a, b)$, donde $f(x) = F(x, \varphi(x))$ y $g(x) = G(x, \varphi(x))$.

1. Calcular los polinomios R_i y τ_i, para $-1 \leq i \leq N$, y ρ_i, para $1 \leq i \leq N + 1$, asociados a F y G como en la Notación 2.24.

2. Determinar las codificaciones de Thom y el orden de todas las raíces $\alpha_1 < \alpha_2 < \cdots < \alpha_k$ en el intervalo (a, b) de los polinomios τ_i, para $-1 \leq i \leq N$, y ρ_i, para $1 \leq i \leq N + 1$.

3. Para todo $0 \leq j \leq k$:

   a) Determinar los signos $\sigma_{I_j,i}$, para $1 \leq i \leq N$, como en la Definición 2.25, para $I_j := (\alpha_j, \alpha_{j+1})$, donde $\alpha_0 = a$ y $\alpha_{k+1} = b$.

   b) Considerar las funciones $p_{I_j,i} := \sigma_{I_j,i} R_i(x, \varphi(x))$. Si $\mathbf{p}_{I_j} = (p_{I_j,i})_{0 \leq i \leq N}$, computar $v_j := v(\mathbf{p}_{I_j}, \alpha_j^+) - v(\mathbf{p}_{I_j}, \alpha_{j+1}^-)$.

4. Para todo $1 \leq j \leq k$, decidir si $f(\alpha_j) = 0$. Si este es el caso, determinar si $g(\alpha_j) > 0$ o $g(\alpha_j) < 0$. Calcular los números:

   - $v^+ := \#\{\ 1 \leq j \leq k\ /\ f(\alpha_j) = 0 \wedge g(\alpha_j) > 0\ \}$
   - $v^- := \#\{\ 1 \leq j \leq k\ /\ f(\alpha_j) = 0 \wedge g(\alpha_j) < 0\ \}$

5. Computar $\mathrm{TaQ}(f, g; a, b) := v^+ - v^- + \sum_{j=0}^{k} v_j$.

Ahora estamos en condiciones de enunciar el siguiente resultado:

Teorema 2.31 *Sea φ una función que satisface la ecuación diferencial $\varphi'(x) = \Phi(x, \varphi(x))$ con $\Phi \in \mathbb{Z}[X, Y]$. Notemos $\delta_X = \deg_X(\Phi)$, $\delta_Y = \deg_Y(\Phi) > 0$, y sean $a < b$ números racionales tales que $[a, b] \subset Dom(\varphi)$. Sean $f(x) = F(x, \varphi(x))$ y $g(x) = G(x, \varphi(x))$ donde $F, G \in \mathbb{Z}[X, Y]$ con $\deg(F) \leq d$, $\deg(G) \leq d$. Existe un algoritmo que calcula $\mathrm{TaQ}(f, g; a, b)$ con complejidad de orden*

$$O\Big(d^4(d+\delta_X+\delta_Y)^3(d+\delta_Y)^3\log^3(d+\delta_X+\delta_Y)\Big)$$

y requiere de $O\big(d^4(d+\delta_X+\delta_Y)^2(d+\delta_Y)\big)$ llamadas al oráculo para determinar el signo en números algebraicos reales de funciones Pfaffianas asociadas a φ definidas por polinomios de grado en X de orden $O(d\delta_X(d+\delta_Y)(d+\delta_X+\delta_Y))$ y grado en Y de orden $O(d\delta_Y(d+\delta_Y)(d+\delta_X+\delta_Y))$.

Demostración. Si $\deg_Y(F), \deg_Y(G) > 0$ se aplica el Algoritmo `Tarski-query`. Analicemos la complejidad de este algoritmo:

1. Por el Teorema 1.42 aplicado a $\widetilde{F}G$ y F con $\mu := (2d + \delta_Y - 1)d + d(2d + \delta_X) = d(4d + \delta_X + \delta_Y - 1)$ (ver cotas para los grados en el Lema 2.23), estos polinomios pueden computarse con una complejidad de orden

$$O(d\big(d+\delta_X+\delta_Y\big)\big((d+\delta_X)(d+\delta_Y)+d^2+d(d+\delta_Y)^{\omega+1}\big)+d^2M(\mu)\log(\mu))\leq$$

$$\leq O\Big(d(d+\delta_X+\delta_Y)(d+\delta_X)(d+\delta_Y)^{\omega+1}+d^3\log^2\big(d+\delta_X+\delta_Y\big)\log\big(\log\big(d+\delta_X+\delta_Y\big)\big)\Big).$$

2. y 3.a) Consideremos el polinomio

$$L = \prod_{-1\leq i\leq N}\tau_i\prod_{1\leq i\leq N+1}\rho_i.$$

Por el Corolario 1.40 y las cotas del Lema 2.23, cada uno de los $2N + 3$ factores tiene grado acotado por el μ del paso 1. Como $N \leq \deg_Y(F)+1 \leq d+1$, obtenemos entonces que

$$\deg(L) \leq (2d + 5)d(4d + \delta_X + \delta_Y - 1) = O(d^2(d + \delta_X + \delta_Y)).$$

Las codificaciones de Thom de las raíces de L en (a, b) ordenadas de menor a mayor y los signos de los polinomios $(\rho_i)_{i=1,\ldots,N+1}$ y $(\tau_i)_{i=-1,\ldots,N}$ entre dos raíces consecutivas de L se calculan simultáneamente buscando las condiciones de signo factibles de las derivadas sucesivas de los polinomios L, $(\rho_i)_{i=1,\ldots,N+1}$ y $(\tau_i)_{i=-1,\ldots,N}$, $X - a$ y $X - b$ sobre los ceros de L (ver Observación 2.12 y Teorema 1.26). Como son $O(\deg(L))$ polinomios en total de grados acotados por $\deg(L)$, por el Teorema 1.22, esto puede hacerse con complejidad de orden

$$\deg(L)\deg(L)^2\log^3(\deg(L)) = O\big(d^6(d+\delta_X+\delta_Y)^3\log^3(d+\delta_X+\delta_Y)\big).$$

De esta forma se obtiene que implementar los pasos del 1 al 3.a) inclusive puede hacerse con complejidad de orden

$$O(d(d+\delta_X+\delta_Y)\left[(d+\delta_X)(d+\delta_Y)^{\omega+1}+d^5(d+\delta_X+\delta_Y)^2\log^3(d+\delta_X+\delta_Y)\right])\leq$$

$$\leq O\Big(d(d+\delta_X+\delta_Y)^3\big((d+\delta_Y)^\omega+d^5\log^3(d+\delta_X+\delta_Y)\big)\Big).$$

3. b) y 4. Estos pasos requieren calcular el signo de funciones Pfaffianas de la forma $H(x,\varphi(x))$, con $H\in\mathbb{Z}[X,Y]$, en números algebraicos (las raíces de L) dados a partir de sus codificaciones de Thom y en los extremos del intervalo (a,b).

   En el paso 3.b), se aplica el Algoritmo `SignAround` (ver Proposición 2.17), para cada función $R_i(x,\varphi(x))$ para $i=0,\ldots,N$, con el polinomio L definido en el paso 2.

   Por la Proposición 1.38 y el Lema 2.23, para $i=0,\ldots,N$, R_i tiene grado en X acotado por $d(4d+\delta_X+\delta_Y-1)$ y grado en Y acotado por d, de donde se obtiene (por la Proposición 2.4) que la multiplicidad de cualquier cero de $R_i(x,\varphi(x))$ está acotada, para todo $i=0,\ldots,N$, por

$$\lambda := d(4d+\delta_X+\delta_Y-1)(2d+\delta_Y-1)+d(\delta_X+1).$$

   Usando que $\deg_X(R_i)+\lambda\delta_X=O(\lambda\delta_X)$ y $\deg_Y(R_i)+\lambda\delta_Y=O(\lambda\delta_Y)$, aplicar $N+1$ veces el algoritmo `SignAround` tiene complejidad

$$O\big(d\lambda M(\lambda\delta_X)M(\lambda\delta_Y)\big)\leq$$

$$\leq O\Big(d^4(d+\delta_X+\delta_Y)^3(d+\delta_Y)^3\log^2(d+\delta_X+\delta_Y)\log^2(\log(d+\delta_X+\delta_Y))\Big)\leq$$

$$\leq O\Big(d^4(d+\delta_X+\delta_Y)^3(d+\delta_Y)^3\log^3(d+\delta_X+\delta_Y)\Big)$$

   y requiere de $O\big(d\lambda\deg(L)\big)\leq O\Big(d^4(d+\delta_X+\delta_Y)^2(d+\delta_Y)\Big)$ llamadas al oráculo para calcular el signo en números reales algebraicos de funciones Pfaffianas asociadas a φ y definidas por polinomios de grado en X acotados por $d(4d+\delta_X+\delta_Y-1)+\lambda\delta_X$ y grado en Y acotado por $d+\lambda(\delta_Y-1)$.

   En el paso 4, hay que hacer a lo sumo $2k$ llamadas más al oráculo, donde $k\leq\deg(L)$. Es decir, $O(d^2(d+\delta_X+\delta_Y))$ llamadas al oráculo para funciones Pfaffianas asociadas a φ definidas por polinomios de grado en X e Y acotados por d.

La complejidad total de este algoritmo es la del paso 3.b).

La correctitud en este caso se sigue del Teorema 2.30, de la Observación 2.28, de la correctitud del algoritmo del Teorema 1.42 y del Algoritmo `SignAround` y del hecho de que el polinomio L definido en el paso 2 tiene como raíces exactamente todas las raíces en el intervalo (a,b) de los polinomios τ_i, para $-1\leq i\leq N$, y ρ_i, para $1\leq i\leq N+1$.

Si $\deg_Y(F)=0$ o $\deg_Y(G)=0$, el número $\mathrm{TaQ}(f,g;a,b)$ puede ser fácilmente calculado dentro del mismo orden de complejidad de la siguiente manera:

- Si $\deg_Y(F) = \deg_Y(G) = 0$, tanto f como g son polinomios. En este caso, aplicar el resultado para polinomios univariados de [3, Theorem 2.70 - Algorithm 8.44].

- Si $\deg_Y(F) = 0$ y $\deg_Y(G) > 0$, usar el oráculo para determinar el signo de g en las raíces de $f \in \mathbb{Z}[X]$.

- Si $\deg_Y(F) > 0$ y $\deg_Y(G) = 0$, alcanza con reemplazar G por el polinomio $G_1(X,Y) = (Y^2+1)G(X,Y)$ pues $\mathrm{TaQ}(f,g;a,b) = \mathrm{TaQ}(f,g_1;a,b)$.

□

2.4. Problema de decisión

En esta sección nos centraremos en el problema de decisión para fórmulas que involucran funciones Pfaffianas de una variable asociadas a una función fija φ que satisface una ecuación diferencial de la forma $\varphi'(x) = \Phi(x, \varphi(x))$, donde $\Phi \in \mathbb{Z}[X,Y]$ con $\deg_Y(\Phi) > 0$.

Siguiendo el enfoque de [16], consideraremos un fragmento de la teoría de primer orden sobre $\mathbb{R}$ aumentada con la función φ; más precisamente, trabajaremos con fórmulas cerradas prenexas construidas a partir de fórmulas atómicas del tipo $f(x) > 0$, $f(x) = 0$ y $f(x) < 0$, donde $f(x) = F(x, \varphi(x))$, con $F \in \mathbb{Z}[X,Y]$, utilizando conjunciones ($\wedge$), disyunciones ($\vee$) y negaciones ($\neg$). Tales fórmulas pueden escribirse en la forma

$$Q\, x \bigvee_i \Big(\bigwedge_j \Psi_{ij}(x) \Big)$$

donde Q es un cuantificador existencial o universal ($\exists$ o $\forall$) y cada $\Psi_{ij}(x)$ es una fórmula atómica.

El problema de decisión que analizaremos consiste en determinar si una fórmula de este tipo es verdadera o falsa sobre $\mathbb{R}$. Presentaremos un procedimiento simbólico para resolver este problema y estimaremos su complejidad, suponiendo que contamos con un oráculo para determinar el signo de funciones de esta clase evaluadas en números reales algebraicos.

Notar que un caso particular es el problema de la consistencia de un sistema de ecuaciones e inecuaciones dadas por funciones Pfaffianas asociadas a una función φ:

$$\exists\, x : f_1(x) \triangle_1 0, \ldots, f_K(x) \triangle_K 0,$$

donde $f_j(x) = F_j(x, \varphi(x))$, con $F_j \in \mathbb{Z}[X,Y]$ y $\triangle_j \in \{<, \leq, =, \neq, \geq, >\}$ para cada j.

Para esto, primero mostraremos un procedimiento para determinar todas las condiciones de signo factibles de una familia finita de funciones Pfaffianas asociadas a φ en un intervalo $[a,b] \subseteq \mathrm{Dom}(\varphi)$, para a y b números algebraicos reales (las condiciones de signo factibles en este caso se definen de la misma manera que para polinomios, ver Definición 1.21).

Para comenzar, dadas funciones Pfaffianas $f, g_1, \ldots, g_s$ asociadas a φ, mostraremos cómo determinar las condiciones de signo factibles de $g_1, \ldots, g_s$ sobre el conjunto

$$Z = \{\ x \in (a,b) \mid f(x) = 0\ \}.$$

Observemos que hay a lo sumo tantas condiciones de signo factibles como elementos haya en Z. Para determinarlas, adaptaremos a nuestro contexto la idea del algoritmo del Teorema 1.22 obteniendo así el siguiente resultado:

Teorema 2.32 *Continuando con la misma notación, si $\delta_X = \deg_X(\Phi)$, $\delta_Y = \deg_Y(\Phi)$, d una cota superior para el grado total de los polinomios que definen a las funciones $f, g_1, \ldots, g_s$ y $\#Z \leq m$ ($m \geq 1$), existe un algoritmo que calcula las condiciones de signo factibles de $g_1, \ldots, g_s$ sobre el conjunto Z con complejidad de orden*

$$O\Big(s\big(md^4 \log^4(m)\big(d\log(m) + \delta_X + \delta_Y\big)^6 \log^3(d\log(m) + \delta_X + \delta_Y) + m^2\big)\Big)$$

y se requiere del orden de $O\Big(smd^4 \log^4(m)\big(d\log(m) + \delta_Y\big)^3\Big)$ llamadas al oráculo para funciones Pfaffianas definidas por polinomios de grado en X de orden $O\big(d\delta_X \log(m)(d\log(m) + \delta_X + \delta_Y)^2\big)$ y grado en Y de orden $O\big(d\delta_Y \log(m)(d\log(m) + \delta_X + \delta_Y)^2\big)$.

Demostración. Nuestro procedimiento es recursivo y, en cada paso $i = 1, \ldots, s$, calcula las condiciones de signo factibles para $g_1, \ldots, g_i$ sobre Z por medio del cálculo de indicadores de Tarski apropiados y de la resolución de un sistema de ecuaciones lineales.

Para $i = 1$, hay tres condiciones de signo posibles:

$$\{x \in Z \mid g_1(x) > 0\},\ \{x \in Z \mid g_1(x) = 0\}\ \ y\ \ \{x \in Z \mid g_1(x) < 0\}.$$

Determinamos los cardinales c^+, c^0 y c^- de estos conjuntos respectivamente, teniendo en cuenta que verifican el siguiente sistema lineal de ecuaciones:

$$\begin{cases} c^0 + c^+ + c^- = \mathrm{TaQ}(f, 1; a, b) \\ \qquad\ c^+ - c^- = \mathrm{TaQ}(f, g_1; a, b) \\ \qquad\ c^+ + c^- = \mathrm{TaQ}(f, g_1^2; a, b) \end{cases}$$

Este paso requiere de calcular 3 indicadores de Tarski de funciones Pfaffianas asociadas a φ definidas por polinomios de grado total a lo sumo $2d$. Por el Teorema 2.31, esto puede hacerse con complejidad de orden

$$O\Big(d^4(d + \delta_X + \delta_Y)^3(d + \delta_Y)^3 \log^3(d + \delta_X + \delta_Y)\Big)$$

y requiere de $O\Big(d^4(d + \delta_X + \delta_Y)^2(d + \delta_Y)\Big)$ llamadas al oráculo para determinar el signo en números algebraicos reales de funciones Pfaffianas asociadas a φ definidas por polinomios de grado en X de orden $O\Big(d\delta_X(d+\delta_Y)(d+\delta_X+\delta_Y)\Big)$ y grado en Y de orden $O\Big(d\delta_Y(d+\delta_Y)(d+\delta_X+\delta_Y)\Big)$.

Para $i > 1$, finalizado el paso $i-1$, cada una de las condiciones de signo factibles calculadas en el paso $i-1$ da lugar a tres posibles condiciones de signo para el paso i. De manera similar al caso de polinomios (ver la demostración de Teorema 1.22), se construye y resuelve un sistema lineal de tamaño a lo sumo $3m \times 3m$ para lo cual se requiere:

- Calcular a lo sumo $3m$ indicadores de Tarski del tipo $\mathrm{TaQ}(f, g_{j_1}^{\alpha_1} \dots g_{j_t}^{\alpha_t}; a, b)$ con $\alpha_j \in \{0, 1, 2\}$, $t \leq \log(m)$. Como los grados de los polinomios que definen las funciones $g_{j_1}^{\alpha_1} \dots g_{j_t}^{\alpha_t}$ están acotados por $2d\log(m)$, la complejidad de calcular cualquiera de estos indicadores de Tarski con el algoritmo del Teorema 2.31 es de orden

$$O(d^4 \log^4(m)\big(d\log(m) + \delta_X + \delta_Y\big)^3\big(d\log(m) + \delta_Y\big)^3 \log^3(d\log(m) + \delta_X + \delta_Y)) \leq$$

$$\leq O\Big(d^4 \log^4(m)\big(d\log(m) + \delta_X + \delta_Y\big)^6 \log^3(d\log(m) + \delta_X + \delta_Y)\Big)$$

y se requiere del orden de

$$O(d^4 \log^4(m)\big(d\log(m) + \delta_Y\big)\big(d\log(m) + \delta_Y + \delta_X\big)^2) \leq O\big(d^4 \log^4(m)\big(d\log(m) + \delta_Y\big)^3\big)$$

llamadas al oráculo para funciones Pfaffianas definidas por polinomios de grado en X de orden

$$O\big(d\delta_X \log(m)\big(d\log(m)+\delta_Y\big)\big(d\log(m)+\delta_Y+\delta_X\big)\big) \leq O(d\delta_X \log(m)(d\log(m)+\delta_X+\delta_Y)^2)$$

y de grado en Y de orden

$$O\big(d\delta_Y \log(m)\big(d\log(m)+\delta_Y\big)\big(d\log(m)+\delta_Y+\delta_X\big)\big) \leq O(d\delta_Y \log(m)(d\log(m)+\delta_X+\delta_Y)^2).$$

- Resolver un sistema lineal de ecuaciones de tamaño a lo sumo $(3m) \times (3m)$ que puede hacerse, según [18], con complejidad de orden $O(m^2)$.

Luego, la complejidad del paso i es de orden:

$$O\Big(md^4 \log^4(m)\big(d\log(m) + \delta_X + \delta_Y\big)^6 \log^3(d\log(m) + \delta_X + \delta_Y) + m^2\Big)$$

y se requiere del orden de $O\Big(md^4 \log^4(m)\big(d\log(m)+\delta_Y\big)^3\Big)$ llamadas al oráculo para funciones Pfaffianas definidas por polinomios de grado en X de orden $O\big(d\delta_X \log(m)(d\log(m) + \delta_X + \delta_Y)^2\big)$ y grado en Y de orden $O\big(d\delta_Y \log(m)(d\log(m) + \delta_X + \delta_Y)^2\big)$.

Se sigue que este procedimiento puede hacerse con una complejidad total de orden

$$O\Big(s\big(md^4 \log^4(m)\big(d\log(m) + \delta_X + \delta_Y\big)^6 \log^3(d\log(m) + \delta_X + \delta_Y) + m^2\big)\Big)$$

y se requiere del orden de $O\Big(smd^4 \log^4(m)\big(d\log(m) + \delta_Y\big)^3\Big)$ llamadas al oráculo para funciones Pfaffianas definidas por polinomios de grado en X de orden $O\big(d\delta_X \log(m)(d\log(m) + \delta_X + \delta_Y)^2\big)$ y grado en Y de orden $O\big(d\delta_Y \log(m)(d\log(m) + \delta_X + \delta_Y)^2\big)$. □

Teniendo en cuenta el hecho de que, por el Corolario 2.15,

$$m = 2d^2(d+1)\big((\delta_Y + 3)d + \delta_X\big) + d$$

es una cota superior de $\#Z$, obtenemos como consecuencia el siguiente resultado:

Corolario 2.33 *Continuando con la misma notación, existe un algoritmo que calcula las condiciones de signo factibles de $g_1, \ldots, g_s$ sobre el conjunto Z con complejidad de orden*

$$O\Big(sd^7(d\delta_Y+\delta_X)\log^4(d\delta_Y+\delta_X)\big(d\log(d\delta_Y+\delta_X)+\delta_X+\delta_Y\big)^6\log^3(d\log(d\delta_Y+\delta_X)+\delta_X+\delta_Y)\Big)$$

y requiere del orden de $O\Big(sd^4\log^4(d\delta_Y+\delta_X)\big(d\log(d\delta_Y+\delta_X)+\delta_Y\big)\big(d\log(d\delta_Y+\delta_X)+\delta_Y+\delta_X\big)^2\big)$ llamadas al oráculo para calcular el signo en números reales algebraicos de funciones Pfaffianas definidas por polinomios de grado en X y de grado en Y de orden $O\Big(d\delta_X\log(d\delta_Y+\delta_X)\big(d\log(d\delta_Y+\delta_X)+\delta_Y\big)\big(d\log(d\delta_Y+\delta_X)+\delta_Y+\delta_X\big)\Big)$ y $O\Big(d\delta_Y\log(d\delta_Y+\delta_X)\big(d\log(d\delta_Y+\delta_X)+\delta_Y\big)\big(d\log(d\delta_Y+\delta_X)+\delta_Y+\delta_X\big)\Big)$ respectivamente.

A continuación, aplicando el Teorema 2.32, determinaremos todas las condiciones de signo factibles de una familia de funciones Pfaffianas $g_1, \ldots, g_s$ asociadas a φ sobre un intervalo $[a, b] \subseteq \mathrm{Dom}(\varphi)$, obteniendo el siguiente resultado:

Teorema 2.34 *Sea φ una función Pfaffiana que satisface una ecuación diferencial de la forma $\varphi'(x) = \Phi(x, \varphi(x))$ en un abierto $\mathcal{U} \subset \mathbb{R}$, con $\Phi \in \mathbb{Z}[X, Y]$ y $\delta_Y = \deg_Y(\Phi) > 0$. Sean $g_1, \ldots, g_s$ funciones Pfaffianas definidas por polinomios de grado total acotado por d. Si $\delta_X = \deg_X(\Phi)$, todas las condiciones de signo factibles para $g_1, \ldots, g_s$ en un intevalo $[a, b] \subseteq Dom(\varphi)$, con a, b números reales algebraicos, pueden ser determinadas con un procedimiento simbólico de complejidad*

$$O(s^2\delta_Y d^6(ds + d\delta_Y + \delta_X)^{14}\log^{11}(ds + d\delta_Y + \delta_X))$$

que efectúa $O(s^2\delta_Y(ds+d\delta_Y+\delta_X)^{11}\log^7(ds+d\delta_Y+\delta_X))$ llamadas al oráculo para determinar el signo en números reales algebraicos de funciones Pfaffianas asociadas a φ definidas por polinomios en $\mathbb{Z}[X, Y]$ de grado en X de orden $O(\delta_X(ds + d\delta_Y + \delta_X)^4\log^2(ds + d\delta_Y + \delta_X))$ y grado en Y de orden $O(\delta_Y(ds + d\delta_Y + \delta_X)^4\log^2(ds + d\delta_Y + \delta_X))$.

Demostración. Supongamos que $g_i(x) = G_i(x, \varphi(x))$ para $i = 1, \ldots, s$, con $G_i \in \mathbb{Z}[X, Y]$. El algoritmo consiste en implementar los siguientes pasos:

1. Para cada $j = 1, \ldots, s$, calcular las condiciones de signo factibles de las funciones $g_1, \ldots, g_{j-1}, g_{j+1}, \ldots, g_s$ sobre los ceros de g_j en el intervalo (a, b).
2. Calcular las condiciones de signo factibles de $g_1, \ldots, g_s$ sobre los ceros de $f = \big(\prod_{1\le j\le s} g_j\big)'$.
3. Utilizar el oráculo para calcular los signos de $g_1, \ldots, g_s$ en a y en b.

La correctitud de este algoritmo se sigue de que en el paso 1, se calculan repitiendo (eventualmente) varias de ellas, todas las condiciones de signo factibles en donde alguna de las funciones es igual a cero. En el paso 2, se calculan todas las condiciones de signo factibles definidas por desigualdades. Observar para ello que las condiciones de signo factibles de $g_1, \ldots, g_s$ sobre los ceros de f incluirán las condiciones de signo factibles de $g_1, \ldots, g_s$ entre dos ceros consecutivos de las funciones g_j, para $1 \le j \le s$, puesto que por el Teorema de Rolle, siempre hay un cero de f entre ellos.

Calculemos la complejidad del algoritmo:

1. Por el Corolario 2.33, este paso puede hacerse con complejidad de orden $O\Big(s^2d^7(d\delta_Y+\delta_X)\log^4(d\delta_Y+\delta_X)\big(d\log(d\delta_Y+\delta_X)+\delta_X+\delta_Y\big)^6\log^3(d\log(d\delta_Y+\delta_X)+\delta_X+\delta_Y)\Big)$ y requiere del orden de $O\Big(s^2d^4\log^4(d\delta_Y+\delta_X)\big(d\log(d\delta_Y+\delta_X)+\delta_Y+\delta_X\big)^3\Big)$ llamadas al oráculo para calcular el signo en números reales algebraicos de funciones Pfaffianas asociadas a φ definidas por polinomios de grado en X y de grado en Y de orden $O\Big(d\delta_X\log(d\delta_Y+\delta_X)\big(d\log(d\delta_Y+\delta_X)+\delta_Y\big)\big(d\log(d\delta_Y+\delta_X)+\delta_Y+\delta_X\big)\Big)$ y $O\Big(d\delta_Y\log(d\delta_Y+\delta_X)\big(d\log(d\delta_Y+\delta_X)+\delta_Y\big)\big(d\log(d\delta_Y+\delta_X)+\delta_Y+\delta_X\big)\Big)$ respectivamente. Usando que $\log^3(d\log(d\delta_Y+\delta_X)+\delta_X+\delta_Y)=O(\log^3(d\delta_Y+\delta_X))$ y que $\big(d\log(d\delta_Y+\delta_X)+\delta_X+\delta_Y\big)^6=O(d^6(d\delta_Y+\delta_X)^6)$, obtenemos que la complejidad de este paso es de orden $O(s^2d^{13}(d\delta_Y+\delta_X)^7\log^7(d\delta_Y+\delta_X))$.

2. Si llamamos $P=\prod_{1\le j\le s}G_j$, el polinomio que define a f es $\frac{\partial P}{\partial X}+\Phi\frac{\partial P}{\partial Y}$, que tiene grado acotado por $sd+\delta_X+\delta_Y-1$. Luego, por el Corolario 2.15, la cantidad de ceros de f es de orden $O\big(\delta_Y(sd+\delta_X+\delta_Y)^4\big)$ y el grado total de todos los polinomios que definen las funciones involucradas está acotado por $sd+\delta_X+\delta_Y$. Aplicando el algoritmo del Teorema 2.32, se tiene que este paso puede hacerse con complejidad de orden

$$O(s\delta_Y(sd+\delta_X+\delta_Y)^{14}\log^{13}(sd+\delta_X+\delta_Y))$$

y requiere del orden de $O\big(s\delta_Y(sd+\delta_X+\delta_Y)^8\log^4(sd+\delta_X+\delta_Y)\big((sd+\delta_X+\delta_Y)\log(sd+\delta_X+\delta_Y)+\delta_X+\delta_Y\big)^3\big)\le O(s\delta_Y(sd+\delta_X+\delta_Y)^{11}\log^7(sd+\delta_X+\delta_Y))$ llamadas al oráculo para funciones Pfaffianas definidas por polinomios de grados en X acotados por $O(\delta_X(sd+\delta_X+\delta_Y)^3\log^3(sd+\delta_X+\delta_Y))$ y grados en Y acotados por $O(\delta_Y(sd+\delta_X+\delta_Y)^3\log^3(sd+\delta_X+\delta_Y))$.

3. Solo requiere de $2s$ llamadas al oráculo para funciones Pfaffianas definidas por polinomios de grado total acotado por d.

La complejidad total de este algoritmo es de orden $O(s^2\delta_Y(ds+d\delta_Y+\delta_X)^{14}\log^7(ds+d\delta_Y+\delta_X)(d^6+\log^6(ds+d\delta_Y+\delta_X)))$ y requiere del orden de $O(s^2\delta_Y(ds+d\delta_Y+\delta_X)^{11}\log^7(ds+d\delta_Y+\delta_X))$ llamadas al oráculo para funciones Pfaffianas definidas por polinomios de grados en X de orden $O(\delta_X(ds+d\delta_Y+\delta_X)^4\log^2(ds+d\delta_Y+\delta_X))$ y grados en Y de orden $O(\delta_Y(ds+d\delta_Y+\delta_X)^4\log^2(ds+d\delta_Y+\delta_X))$. □

Observación 2.35 *Si se tienen en cuenta las cotas para la cantidad de ceros de $g_1,\dots,g_s$ y $(\prod_{1\le i\le s}g_i)'$ (ver Corolario 2.15), la cantidad de condiciones de signo factibles de las funciones $g_1,\dots,g_s$ en un intervalo contenido en Dom(φ) está acotada por $O(\delta_Y(sd+\delta_X+\delta_Y)^4)$.*

Finalmente, como consecuencia del Teorema 2.34 podemos establecer un resultado de complejidad para el problema de decisión formulado al comienzo de la sección.

Corolario 2.36 *Sea Ψ una fórmula prenexa en una variable x que involucra funciones Pfaffianas $g_1,\dots,g_s$ definidas por $g_i(x)=G_i(x,\varphi(x))$ para $G_i\in\mathbb{Z}[X,Y]$ con $\deg(G_i)\le d$*

$(1 \leq i \leq s)$ y una función Pfaffiana φ que satisface la ecuación $\varphi'(x) = \Phi(x, \varphi(x))$ para $\Phi \in \mathbb{Z}[X, Y]$ con $\deg_X(\Phi) = \delta_X$ y $\deg_Y(\Phi) = \delta_Y > 0$. El valor de verdad de Ψ en un intervalo $[a, b] \subseteq Dom(\varphi)$, para a y b números reales algebraicos, puede ser determinado con complejidad $O(s^2\delta_Y(ds+d\delta_Y+\delta_X)^{14}\log^7(ds+d\delta_Y+\delta_X)(d^6+\log^6(ds+d\delta_Y+\delta_X))+\delta_Y(sd+\delta_X+\delta_Y)^4|\Psi|)$, donde $|\Psi|$ denota la longitud de Ψ.

Capítulo 3

Un caso particular: E-polinomios

En este capítulo trabajaremos con una familia particular de funciones Pfaffianas de orden 1 que involucran a funciones exponenciales, llamadas E-polinomios, para las cuales es posible determinar de manera efectiva el signo que toman al ser evaluadas en números reales algebraicos.

En la Sección 3.1, las definiremos y analizaremos algunas de sus propiedades. En la Sección 3.2, construiremos un algoritmo que permite calcular el signo de un E-polinomio en un número real algebraico dado por su codificación de Thom como raíz de un polinomio sin necesidad de recurrir a un oráculo. Este hecho central nos permitirá diseñar algoritmos efectivos para calcular la cantidad de ceros de un E-polinomio en un intervalo, que mostraremos en la Sección 3.3. En la Sección 3.4, resolveremos el problema de decisión descripto en el capítulo anterior para estas funciones, calculando además las complejidades algebraicas de los algoritmos diseñados, y veremos también una breve aplicación a E-polinomios de varias variables. En la Sección 3.5, mostraremos algunas generalizaciones de resultados válidos para polinomios a E-polinomios: una cota superior del tamaño de los ceros de un E-polinomio, la noción de codificación de Thom de un cero de un E-polinomio junto con un algoritmo para calcularla, y una generalización del Teorema de Budan-Fourier para la obtención de una cota superior para la cantidad de ceros de un E-polinomio.

3.1. Definición y propiedades básicas

Definición 3.1 *Un* E-polinomio en una variable con coeficientes en $\mathbb{Z}$ *es una función de la forma* $f(x) = F(x, e^{h(x)})$*, donde* $F(X, Y) \in \mathbb{Z}[X, Y]$ *y* $h \in \mathbb{Z}[X]$ *es no constante.*

Notar que como la función $\varphi(x) = e^{h(x)}$ es solución de la ecuación diferencial $\varphi'(x) = \Phi(x, \varphi(x))$, donde

$$\Phi(X, Y) = h'(X)Y \in \mathbb{Z}[X, Y],$$

se tiene que $f(x) = F(x, e^{h(x)})$ resulta ser una función Pfaffiana de orden 1 asociada a la función $\varphi(x) = e^{h(x)}$.

La clase de los E-polinomios para un polinomio h fijo contiene a los polinomios de $\mathbb{Z}[X]$, y es claro que resulta ser cerrada para la suma, el producto y la derivación.

Continuando con la misma notación, dada $f(x) = F(x, e^{h(x)})$, definimos como en (2.1) pero para el caso particular $\varphi(x) = e^{h(x)}$, el polinomio

$$\widetilde{F}(X,Y) = \frac{\partial F}{\partial X}(X,Y) + h'(X)Y\frac{\partial F}{\partial Y}(X,Y), \tag{3.1}$$

de forma que

$$f'(x) = \widetilde{F}(x, e^{h(x)}).$$

El siguiente resultado nos muestra cotas superiores para los grados en X y en Y y las alturas de los polinomios que definen a f y las derivadas sucesivas de f, en función de las cotas correspondientes al polinomio que define a f.

Lema 3.2 *Con la notación anterior, si* $\deg_X(F) = d_X$, $\deg_Y(F) = d_Y$ *y* $\deg(h) = \delta$ *entonces:*

1) $\deg_X(\widetilde{F}) \leq \delta - 1 + d_X$ *y* $\deg_Y(\widetilde{F}) = d_Y$.

2) $H(\widetilde{F}) \leq H(F)(d_X + d_Y\delta^2 H(h))$.

3) *Si* $f^{(k)}(x) = F_k(x, e^{h(x)})$, *con* $F_k \in \mathbb{Z}[X,Y]$, *se tiene que*

   a) $\deg_X(F_k) \leq k(\delta-1) + d_X, \quad \deg_Y(F_k) = d_Y$ *y*

   b) $H(F_k) \leq H(F)\prod_{j=0}^{k-1}(j(\delta-1) + d_X + d_Y\delta^2 H(h))$.

Demostración.

1) Aplicando el Lema 2.3 a f, como $\delta_X = \delta - 1$ y $d_Y = 1$, se obtiene que $\deg_X(\widetilde{F}) \leq \delta - 1 + d_X$ y que $\deg_Y(\widetilde{F}) \leq d_Y$. Con respecto al grado en Y, si llamamos $a(X)$ al coeficiente de Y^{d_Y} de F, el coeficiente de Y^{d_Y} de $\widetilde{F}$ es $a'(X) + h'(X)d_Y a(X) \neq 0$. Luego, $\deg_Y(\widetilde{F}) = d_Y$.

2) Por la Proposición 1.11, $H(\widetilde{F}) \leq d_X H(F) + H\big(h'(X)Y\frac{\partial F}{\partial Y}\big)$. Aplicando además la desigualdad (1.5), se obtiene que $H\big(h'(X)Y\frac{\partial F}{\partial Y}\big) \leq (\text{mín}\,\{\delta-1, d_X\}+1)d_Y\delta H(h)H(F) \leq d_Y\delta^2 H(h)H(F)$. Luego, $H(\widetilde{F}) \leq H(F)(d_X + d_Y\delta^2 H(h))$ como se quería probar.

3) Ambos ítems se obtienen si hacemos inducción en k, notando que si $k = 1$, $F_1 = \widetilde{F}$, y que $F_{k+1} = \widetilde{F_k}$. La cota para los grados es un caso particular de la demostrada en el Lema 2.3, mientras que para la de la altura basta observar que

$$H(F_{k+1}) = H(\widetilde{F_k}) \leq H(F_k)(\deg_X(F_k) + \deg_Y(F_k)\delta^2 H(h)) \leq$$

$$\leq H(F)\prod_{j=0}^{k-1}\big(j(\delta-1) + d_X + d_Y\delta^2 H(h)\big)(k(\delta-1) + d_X + d_Y\delta^2 H(h)) =$$

$$= H(F)\prod_{j=0}^{k}\big(j(\delta-1) + d_X + d_Y\delta^2 H(h)\big).$$

□

Por otro lado, también obtenemos una cota específica para la multiplicidad de un cero de un E-polinomio como consecuencia de la Proposición 2.4, siendo que $\delta_X = \deg(h) - 1$ y $\delta_Y = 1$:

Proposición 3.3 *Sean $F \in \mathbb{Q}[X,Y]$ con $\deg_Y(F) > 0$ y $h \in \mathbb{Q}[X]$ no constante. Si $\alpha \in \mathbb{R}$ es un cero del E-polinomio $f(x) = F(x, e^{h(x)})$, entonces*

$$\operatorname{mult}(\alpha, f) \leq 2\deg_X(F)\deg_Y(F) + \deg(h)\deg_Y(F).$$

3.2. Determinación de signo de un E-polinomio

En esta sección, construiremos un algoritmo simbólico para determinar el signo de un E-polinomio evaluado en un número real algebraico sobre $\mathbb{Q}$ dado por su codificación de Thom.

Como trabajaremos con funciones exponenciales comenzaremos con algunos resultados previos que nos permitirán manipularlas. Primero, veremos una condición para que un número real algebraico sea raíz de un E-polinomio. Luego, veremos herramientas algorítmicas para aproximar por números racionales el número e^α, con α un número real algebraico sobre $\mathbb{Q}$. Finalmente, daremos dos subrutinas que utilizaremos en nuestro algoritmo principal.

Lema 3.4 *Sea α un número real algebraico y sean $F \in \mathbb{Q}[X,Y]$, con $\deg_Y(F) > 0$, sin factor en $\mathbb{Q}[X]$ de grado positivo, y $h \in \mathbb{Q}[X]$ no nulo. Entonces*

$$F(\alpha, e^{h(\alpha)}) = 0 \Longrightarrow h(\alpha) = 0.$$

Demostración. Es una aplicación directa del Teorema de Lindemann, que afirma que si $\beta \neq 0$ es un número algebraico sobre $\mathbb{Q}$, entonces e^β es trascendente. □

La siguiente cota inferior será útil para el desarrollo del algoritmo que construiremos.

Teorema 3.5 *(ver [28, Section 3]) Sean α, β números algebraicos sobre $\mathbb{Q}$, no nulos. Supongamos que $\deg(\alpha), \deg(\beta) < d$ y que sus alturas están acotadas superiormente por τ. Entonces*

$$\left|e^\beta - \alpha\right| > e^{-2^{42}d^6(\ln(\tau+e^e))(\ln(\tau)+\ln(\ln\tau))}.$$

En el siguiente lema, mostraremos dos algoritmos que utilizaremos como herramientas para aproximar e^α, con α un número real algebraico sobre $\mathbb{Q}$.

Lema 3.6 *Sean $P \in \mathbb{Z}[X]$ de grado d y altura H, $\alpha \in \mathbb{R}$ una raíz de P y $\theta \in (0,1)$ un número racional. Entonces, dada la codificación de Thom de α como raíz de P, $\sigma_P(\alpha)$:*

i) Existe un algoritmo que encuentra $w_1 \in \mathbb{Q}$ tal que $|\alpha - w_1| < \theta$ en tiempo

$$O\big(d^3\log(\frac{H}{\theta}) + d^3\log^3(d)\big).$$

ii) Existe un algoritmo que encuentra $w_2 \in \mathbb{Q}$ tal que $|e^\alpha - w_2| < \theta$ en tiempo

$$O\big(d^3(\log(\frac{H}{\theta}) + \log^3(d) + H)\big).$$

Demostración.

i) Si aplicamos el algoritmo de la Proposición 1.20 a P y θ se obtienen intervalos $(a, b]$ con $a, b \in \mathbb{Q}$, disjuntos, de longitud menor que θ, tales que todos contienen alguna raíz real de P y todas ellas están contenidas en alguno. Para hallar el intervalo que contiene a α basta aplicar el algoritmo del Teorema 1.22 a los polinomios P, $P_i = P^{(i)}$ para $i = 1, \ldots, d$, $P_a(X) = X - a$ y $P_b(X) = X - b$ para cada a, b extremos de los intervalos calculados antes. Se tiene que α pertenece al intervalo $(a, b]$ tal que la condición de signo $P = 0$, $P_i = (\sigma_P(\alpha))_i$, para $i = 1, \ldots, d$, $P_a > 0$ y $P_b \epsilon 0$ con $\epsilon \in \{<, =\}$ es factible.

Si $\alpha \in (a, b]$, tomar $w_1 = b$.

Para analizar la complejidad, observar que aplicar el algoritmo de la Proposición 1.20 y el algoritmo del Teorema 1.22, nos da una complejidad de $O(d^3 \log(\frac{H}{\theta}) + d^3 \log^3(d))$.

ii) La demostración de este resultado se puede realizar en forma similar a la de [27, Lemma 14].

- Utilizando el algoritmo de la parte *i)* hallamos $w_1 \in \mathbb{Q}$ tal que $|\alpha - w_1| < \widetilde{\theta}$, con $\widetilde{\theta} := \frac{\theta}{2,3^{2+H}}$. Por el Teorema del valor medio se tiene que existe w entre α y w_1 tal que $|e^\alpha - e^{w_1}| = e^w \, |\alpha - w_1|$. Como $|w - \alpha| < |\alpha - w_1| \leq \widetilde{\theta} < 1$, por el Lema 1.16, se tiene que $w < 2 + H$. Luego, $|e^\alpha - e^{w_1}| < e^{2+H}\widetilde{\theta} < \frac{\theta}{2}$.
  Como $\widetilde{\theta} = O(\theta 3^{-(H+2)})$, $d^3 \log(\frac{H}{\widetilde{\theta}}) + d^3 \log^3(d) = O\big(d^3(H + \log(\frac{H}{\theta}) + \log^3(d))\big)$, que resulta ser la complejidad de este paso.

- Sea $t := \big\lceil 9(|w_1| - \ln(\frac{\theta}{2}))\big\rceil - 1 > 0$. Evaluamos el polinomio $p_t(X) = \sum_{i=0}^{t} \frac{X^i}{i!}$ en w_1 y definimos $w_2 := p_t(w_1)$.
  Por la fórmula de Lagrange del resto del Polinomio de Taylor, se tiene que

  $$|e^{w_1} - w_2| \leq \frac{|w_1|^{t+1}}{(t+1)!} e^{|w_1|}.$$

  Renombrando $a = |w_1|$, $b = t+1$ y $c = \frac{\theta}{2}$, veamos que $\frac{a^b}{b!} e^a < c$, o equivalentemente, $a - \ln(c) < \ln(b!) - b \ln(a)$. Utilizando la desigualdad de Stirling $\ln(b!) \geq b \ln(b) - b$ y la desigualdad $b \geq 9 \, |w_1| \geq ae^2$, se obtiene

  $$\ln(b!) - b \ln(a) \geq b \ln(b) - b - b \ln(a) = b(\ln(\frac{b}{a}) - 1) \geq b \geq 9(a - \ln(c)) \geq a - \ln(c),$$

  como se quería probar.
  La complejidad de este paso es $O(t) = O(H + \log(\frac{1}{\theta}))$, pues $|w_1| \leq H + 2$.

Finalmente, se verifica que $|e^\alpha - w_2| \leq |e^\alpha - e^{w_1}| + |e^{w_1} - w_2| < \frac{\theta}{2} + \frac{\theta}{2} = \theta$.
La complejidad total de este algoritmo es de orden $O\big(d^3(H + \log(\frac{H}{\theta}) + \log^3(d))\big)$.

□

A continuación, describiremos una de las subrutinas que utilizaremos en el algoritmo para el cálculo del signo de un E-polinomio en un número algebraico. Con ella, podremos determinar el signo de una expresión de la forma $e^{\beta}-\alpha$, para α y β números reales algebraicos sobre $\mathbb{Q}$.

Algoritmo `SignExpAlg`

INPUT: Números reales algebraicos sobre $\mathbb{Q}$, α y β, dados por su codificación de Thom $\sigma_{P_1}(\alpha)$ y $\sigma_{P_2}(\beta)$ con respecto a los polinomios $P_1, P_2 \in \mathbb{Z}[X]$ tales que $\deg(P_1), \deg(P_2) \leq d$ $(d \geq 2)$ y $H(P_1), H(P_2) \leq H$.

OUTPUT: El signo $s := \operatorname{sg}(e^{\beta} - \alpha)$.

1. Tomar $c := (2^{d+1}(d+1)H)^{-2^{41}d^6(5d+4\lceil \log(H)\rceil)}$.
2. Calcular $w \in \mathbb{Q}$ tal que $|e^{\beta} - w| < \frac{c}{4}$.
3. Calcular $w' \in \mathbb{Q}$ tal que $|\alpha - w'| < \frac{c}{4}$.
4. Calcular $s = \operatorname{sg}(w - w')$.

A partir del algoritmo anterior podemos enunciar el siguiente resultado:

Proposición 3.7 *Sean α y β números reales algebraicos sobre $\mathbb{Q}$ no nulos y $P_1, P_2 \in \mathbb{Z}[X]$, de grados d_1, d_2 respectivamente, tales que $P_1(\alpha) = 0$ y $P_2(\beta) = 0$. Dadas las codificaciones de Thom $\sigma_{P_1}(\alpha), \sigma_{P_2}(\beta)$, el Algoritmo `SignExpAlg`, calcula el signo de $e^{\beta} - \alpha$ en tiempo*

$$O\Big(d^9(d + \log(H))^2 + d^3H\Big),$$

donde $d = \max\{2, d_1, d_2\}$ y $H = \max\{H(P_1), H(P_2)\}$.

Demostración. Analicemos la complejidad de este algoritmo:

En el paso 2, se aplica el algoritmo del Lema 3.6 *ii)* a β con $\theta = \frac{c}{4}$ para obtener $w \in \mathbb{Q}$ tal que $|e^{\beta} - w| < \frac{c}{4}$. Este paso puede realizarse con $O\big(d^3 \log(\frac{H}{c}) + d^3 \log^3(d) + H\big)$ operaciones.

En el paso 3, se utiliza el algoritmo del Lema 3.6 *i)* aplicado a α con $\theta = \frac{c}{4}$ y obtener $w' \in \mathbb{Q}$ tal que $|\alpha - w'| < \frac{c}{4}$. Este paso puede realizarse con complejidad de $O\big(d^3 \log(\frac{H}{c}) + d^3 \log^3(d)\big)$.

Usando que $\log(\frac{H}{c}) = O\big(d^6(d + \log(H))^2\big)$, se obtiene una complejidad total de orden $O\big(d^9(d + \log(H))^2 + d^3H\big)$.

Para el análisis de la correctitud:

- Veamos que se verifica $|e^{\beta} - \alpha| > c$.
  Por el Teorema 3.5, si α y β son números reales algebraicos de grados acotados por d y alturas acotadas por ν, se tiene que

$$|e^{\beta} - \alpha| > e^{-2^{42}d^6 \ln(\nu + e^e)(\ln(\nu) + \ln\ln(\nu))}.$$

Notando que

$$e^{2^{42}d^6\ln(\nu+e^e)(\ln(\nu)+\ln\ln(\nu))} \leq (\nu+16)^{2^{42}d^6(\ln(\nu)+\ln\ln(\nu))} \leq (\nu+16)^{2^{43}d^6\ln(\nu)},$$

tomando $\nu = 2^d(d+1)^{1/2}H$ (ver el Lema 1.10) y usando las cotas

$$2^d(d+1)^{1/2}H + 16 \leq 2^{d+1}(d+1)H \quad \text{y} \quad \ln(2^d(d+1)^{1/2}H) \leq \frac{5}{4}d + \lceil \log(H) \rceil,$$

se obtiene la cota propuesta.

- Veamos que $\mathrm{sg}(e^\beta - \alpha) = \mathrm{sg}(w - w')$.

  - Si $e^\beta < \alpha$: Se tiene que $0 < c < \alpha - e^\beta$. Luego,

$$\omega' - \omega = \omega' - \alpha + \alpha - e^\beta + e^\beta - \omega > -\frac{c}{4} + c - \frac{c}{4} > 0.$$

  - Si $e^\beta > \alpha$: Se tiene que $0 < c < e^\beta - \alpha$. Luego,

$$\omega - \omega' = \omega - \alpha + \alpha - e^\beta + e^\beta - \omega' > -\frac{c}{4} + c - \frac{c}{4} > 0.$$

□

Otra subrutina que utilizaremos nos permitirá localizar un número real de la forma $e^{h(\alpha)}$, para un número algebraico α, entre dos raíces reales consecutivas de un polinomio dado. La describimos a continuación:

Algorithm `RootBox`

INPUT: Un polinomio $h \in \mathbb{Z}[X]$, α un número real algebraico sobre $\mathbb{Q}$, con $h(\alpha) \neq 0$, dado por su codificación de Thom como raíz de un polinomio $L \in \mathbb{Z}[X]$, y un polinomio $M \in \mathbb{Z}[X]$ junto con la lista de las codificaciones de Thom de todas sus raíces reales $\lambda_1 < \lambda_2 < \cdots < \lambda_m$.

OUTPUT: El índice $0 \leq i_0 \leq m$, tal que $\lambda_{i_0} < e^{h(\alpha)} < \lambda_{i_0+1}$, donde $\lambda_0 = -\infty$ y $\lambda_{m+1} = +\infty$.

1. Calcular $S(T) := \mathrm{Res}_X(L(X), T - h(X))$.
2. Calcular las condiciones de signo factibles de los polinomios $L^{(j)}$, $S^{(i)}(h)$, para $0 \leq j \leq \deg(L)$, $0 \leq i \leq \deg(S)$ y determinar, a partir de ellas, la codificación de Thom de $h(\alpha)$ como raíz de S.
3. Calcular $\mathrm{sg}(e^{h(\alpha)} - \lambda_i)$ aplicando el Algoritmo `SignExpAlg`, para $i = 1, \ldots, m$. Llamar i_0+1 al primer índice con el que se obtenga un signo negativo. Si todos fuesen positivos, $i_0 = m$.

A partir de este algoritmo, podemos enunciar el siguiente resultado:

Proposición 3.8 *Sean $h \in \mathbb{Z}[X]$ y $\alpha \in \mathbb{R}$ un número algebraico tal que $h(\alpha) \neq 0$, dado por su codificación de Thom como raíz de un polinomio $L \in \mathbb{Z}[X]$. Sea $M \in \mathbb{Z}[X]$ y $\lambda_1 < \lambda_2 < \ldots < \lambda_m$ sus raíces reales, dadas a partir de sus codificaciones de Thom. El Algoritmo* `RootBox` *calcula el índice i_0, $0 \leq i_0 \leq m$, tal que $\lambda_{i_0} < e^{h(\alpha)} < \lambda_{i_0+1}$, donde $\lambda_0 = -\infty$ y $\lambda_{m+1} = +\infty$, en tiempo*

$$O\Big(m\,\text{máx}\{\eta,\ell\}^3\Big(\mathcal{H} + \text{máx}\{\eta,\ell\}^6(\text{máx}\{\eta,\ell\} + \log(\mathcal{H}))^2\Big)\Big),$$

donde $\deg(L) \leq \ell$, $\deg(h) \leq \delta$, $\deg(M) \leq \eta$ *y* $\mathcal{H} := \text{máx}\{H(M), (\ell+\delta)!\,H(L)^\delta(2H(h))^\ell\}$.

Demostración. Analicemos la complejidad de este algoritmo:

1. Calculamos el polinomio $S(T) \in \mathbb{Z}[X]$, utilizando el Lema 1.30. Esto puede hacerse con $O(\ell(\ell+\delta)^\omega)$ operaciones. Notar que $\deg(S) \leq \ell$.

2. Aplicamos el algoritmo del Teorema 1.22 a $2\ell+2$ polinomios de grados acotados por $\ell\delta$. Esto puede hacerse con complejidad de orden $O(\ell^3\delta^2\log^3(\ell\delta))$.

3. Por la Observación 1.39 se tiene que $H(S) \leq (\ell+\delta)!\,H(L)^\delta(2H(h))^\ell$ y, por lo tanto, $\text{máx}\,\{H(S), H(M)\} \leq \mathcal{H}$. Teniendo en cuenta que debe aplicarse a lo sumo m veces el Algoritmo `SignExpAlg` y que $m\big(\text{máx}\{\eta,\ell\}^9(\text{máx}\{\eta,\ell\} + \log(\mathcal{H}))^2 + \text{máx}\{\eta,\ell\}^3\mathcal{H}\big) = O\Big(m\,\text{máx}\{\eta,\ell\}^3\Big(\mathcal{H} + \text{máx}\{\eta,\ell\}^6(\text{máx}\{\eta,\ell\} + \log(\mathcal{H}))^2\Big)\Big)$, esta última es la complejidad de este paso.

La complejidad total del algoritmo es del orden de la complejidad del paso 3).

Para el análisis de la correctitud, hay que observar que $h(\alpha)$ es raíz de $S(T)$ y que las condiciones factibles determinadas en el paso 2 tienen, para cada raíz ξ de L, en las primeras coordenadas la codificación de Thom de ξ como raíz de L y en la últimas, la de $h(\xi)$ como raíz de S. La correctitud se sigue de la correctitud del Algoritmo `SignExpAlg`. □

Ahora estamos en condiciones de introducir el algoritmo principal de esta sección: un algoritmo que, dada la codificación de Thom de un número real algebraico sobre $\mathbb{Q}$, calcula el signo de un E-polinomio evaluado en ese número. Este algoritmo reemplazará al oráculo utilizado en el Capítulo 2.

Algoritmo `E-SignDetermination`

INPUT: Polinomios $F \in \mathbb{Z}[X,Y]$, $h \in \mathbb{Z}[X]$, $\deg(h) > 0$, $L \in \mathbb{Z}[X]$ y la lista ordenada de las codificaciones de Thom $\sigma_L(\alpha_1), \ldots, \sigma_L(\alpha_t)$ de las raíces reales $\alpha_1, \ldots, \alpha_t$ de L.

OUTPUT: Los signos $\sigma_j := F(\alpha_j, e^{h(\alpha_j)})$ para $1 \leq j \leq t$.

1. Para todo $1 \leq j \leq t$, determinar si $F(\alpha_j, Y) \equiv 0$.
   Si este es el caso, el signo de $F(\alpha_j, e^{h(\alpha_j)})$ es 0.

2. Calcular $R = \gcd(L, h)$ y la lista de las condiciones de signo factibles de los polinomios $L, L^{(1)}, \ldots, L^{(\deg(L))}, R, F(X,1)$. Para todo j tal que $F(\alpha_j, Y) \not\equiv 0$ y $R(\alpha_j) = 0$, recorriendo la lista anterior, determinar los signos de $F(\alpha_j, e^{h(\alpha_j)}) = F(\alpha_j, 1)$.

3. Calcular $M(Y) := \mathrm{Res}_X(L(X), F(X,Y))$.

4. Calcular la lista ordenada de las codificaciones de Thom de las raíces reales de M: $\lambda_1 < \ldots < \lambda_m$.

5. Para todo $1 \leq j \leq t$ tal que $F(\alpha_j, Y) \not\equiv 0$ y $R(\alpha_j) \neq 0$:

   a) Determinar el índice $0 \leq i_j \leq m$ tal que $\lambda_{i_j} < e^{h(\alpha_j)} < \lambda_{i_j+1}$ aplicando la subrutina `RootBox`, donde $\lambda_0 := -\infty$ y $\lambda_{m+1} := +\infty$.

   b) Encontrar $w_j \in \mathbb{Q} \cap (\lambda_{i_j}, \lambda_{i_j+1})$.

   c) Calcular el signo del polinomio $g_j(X) := F(X, w_j) \in \mathbb{Q}[X]$ en $X = \alpha_j$. Llamar $\sigma_j := \mathrm{sg}\big(g_j(\alpha_j)\big)$

A continuación, enunciamos el resultado más importante de esta sección.

Teorema 3.9 *Sean $F \in \mathbb{Z}[X,Y]$, $h, L \in \mathbb{Z}[X]$ y $\alpha_1, \ldots, \alpha_t$ raíces reales de L dadas por sus codificaciones de Thom $\sigma_L(\alpha_1), \ldots, \sigma_L(\alpha_t)$. El Algoritmo* `E-SignDetermination` *calcula el signo de $F(\alpha_j, e^{h(\alpha_j)})$, para $j = 1, \ldots, t$, en tiempo*

$$O\Big(t(\ell d_Y)^4(\mathcal{H} + (\ell d_Y)^6(\ell d_Y + \log(\mathcal{H}))^2)\Big),$$

donde $\mathcal{H} = \max\{(\ell+\delta)!H(L)^\delta(2H(h))^\ell, (\ell+d_X)!H(L)^{d_X}((d_Y+1)H(F))^\ell\}$, $\deg_X(F) \leq d_X$, $\deg_Y(F) \leq d_Y$, $\deg(L) \leq \ell$ y $\deg(h) \leq \delta$.

Demostración. Analicemos primero la correctitud de este algoritmo:

Por el Lema 3.4, se tiene que para un número α, real y algebraico sobre $\mathbb{Q}$,

$$F(\alpha, e^{h(\alpha)}) = 0 \iff F(\alpha, Y) \equiv 0 \text{ o } \Big(h(\alpha) = 0 \text{ y } F(\alpha, 1) = 0\Big). \tag{3.2}$$

En consecuencia, los pasos 1) y 2) permiten determinar los índices j tales que $F(\alpha_j, e^{h(\alpha_j)}) = 0$. Observar que, si α es raíz de L, entonces $\mathrm{cont}(F)(\alpha) = 0$ si y sólo si $(\sigma_L(\alpha), 0)$ da una condición de signo factible para $L', \ldots, L^{(\deg(L))}, \mathrm{cont}(F)$ sobre los ceros de L, donde $\mathrm{cont}(F)$ es el gcd de los coeficientes de F como polinomio en la variable Y.

Veamos que el signo de cada polinomio $F(X, w_j)$ en $X = \alpha_j$ es el signo de $F(\alpha_j, e^{h(\alpha_j)})$.

Como α_j no es raíz de R, se tiene que $e^{h(\alpha_j)} \neq 1$. Luego, $e^{h(\alpha_j)} - \lambda_i \neq 0 \ \ \forall i = 1, \ldots, m$, pues si no $e^{h(\alpha_j)}$ resultaría ser algebraico sobre $\mathbb{Q}$, por lo que $h(\alpha_j) = 0$. Como M no cambia de signo en $(\lambda_{i_j}, \lambda_{i_j+1})$, tampoco lo hace $F(\alpha_j, Y) \in \mathbb{R}[Y]$ (pues si existiera $y \in \mathbb{R}$ tal que $F(\alpha_j, y) = 0$, y sería raíz de M). Luego $\forall w \in \mathbb{Q} \cap (\lambda_{i_j}, \lambda_{i_j+1})$ se tiene que

$$\mathrm{sg}\Big(F(\alpha_j, e^{h(\alpha_j)})\Big) = \mathrm{sg}\Big(F(\alpha_j, w)\Big) = \mathrm{sg}\Big(F(X, w)\mid_{X=\alpha_j}\Big).$$

Para el análisis de la complejidad, se debe observar que:

1. - Utilizar el algoritmo rápido de Euclides para calcular $\mathrm{cont}(F)$(ver Sección 1.2). Esto puede hacerse con complejidad $O(d_Y M(d_X)\log(d_X)) = O(d_Y d_X^2)$.

- Para decidir si $F(\alpha_j, Y) \equiv 0$, calculamos las condiciones de signo factibles de las derivadas de L y de $\mathrm{cont}(F)$ sobre el conjunto de ceros de L con complejidad $O\big(\ell^2 \operatorname{máx}\{\ell, d_X\} \log(\ell) \log^2(\operatorname{máx}\{\ell, d_X\})\big)$(ver Teorema 1.22).

2. - Se puede calcular R con complejidad de $O(M(\operatorname{máx}\{\ell, \delta\}) \log(\operatorname{máx}\{\ell, \delta\}))$ (ver Sección 1.2).
   - Las condiciones de signo factibles de las derivadas de L y de $F(X, 1)$ sobre $R = 0$ se pueden calcular con complejidad $O\big(\ell^2 \operatorname{máx}\{\ell, d_X\} \log(\ell) \log^2(\operatorname{máx}\{\ell, d_X\})\big)$ (ver Teorema 1.22).

3. Utilizando el algoritmo de la Proposición 1.30, $M(Y)$ puede calcularse con complejidad de orden $O(\ell d_Y (d_X + \ell)^\omega + M(\ell d_Y) \log(\ell d_Y))$.

4. Como $\deg(M) \leq \ell d_Y$, puede calcularse la lista ordenada de las codificaciones de Thom de las raíces reales de M con complejidad $O((\ell d_Y)^3 \log^3(\ell d_Y))$.

5. *a*) Notando que, por la Observación 1.39, $H(M) \leq (\ell + d_X)!((d_Y + 1)H(F))^\ell H(L)^{d_X}$ y que $\ell d_Y = \operatorname{máx}\{\ell, \ell d_Y\}$, la complejidad de aplicar el algoritmo `RootBox` (ver Proposición 3.8) es de orden

$$O\big((\ell d_Y)^4 (\mathcal{H} + (\ell d_Y)^6 (\ell d_Y + \log(\mathcal{H}))^2)\big),$$

donde $\mathcal{H} := \operatorname{máx}\{(\ell + d_X)!((d_Y + 1)H(F))^\ell H(L)^{d_X}, (\ell + \delta)!\, H(L)^\delta (2H(h))^\ell\}$.

b) Dado que una cota inferior de la distancia mínima entre dos raíces reales distintas de M es $\varepsilon = (\ell d_Y)^{-\frac{\ell d_Y + 2}{2}} (\ell d_Y + 1)^{\frac{1 - \ell d_Y}{2}} ((\ell + d_X)!\, H(L)^{d_X} ((d_Y + 1)H(F))^\ell)^{1 - \ell d_Y}$ (por la desigualdad (1.6)), utilizando el algoritmo de la Proposición 1.20 aplicado al polinomio M y a ε, obtenemos intervalos disjuntos $(a_i, b_i]$ con extremos racionales, tales que $\lambda_i \in (a_i, b_i]$ para $i = 1, \ldots, m$. Tomar $w_j := b_{i_j}$.
Dado que $\deg^3(M) \log(\frac{H(M)}{\varepsilon}) = O\big((\ell d_Y)^3 \log\big((\ell d_Y)^{\frac{\ell d_Y + 2}{2}} (\ell d_Y + 1)^{\frac{\ell d_Y - 1}{2}} ((\ell + d_X)! \times H(L)^{d_X} ((d_Y + 1)H(F))^\ell)^{\ell d_Y}\big)\big)$ y teniendo en cuenta que $\log((\ell + d_X)!) = O((\ell + d_X) \log(\ell + d_X))$, se tiene que este paso puede hacerse con complejidad

$$O\Big((\ell d_Y)^4 \big((\ell + d_X) \log(\ell + d_X) + \ell(\log(H(F)) + \log(d_Y)) + d_X \log(H(L))\big)\Big).$$

c) - Calcular los coeficientes de $g_j(X) := F(X, w_j)$ requiere de evaluar a lo sumo d_X polinomios de una variable, de grado a lo sumo d_Y, en un número racional. Esto puede hacerse con complejidad $O(d_X d_Y)$.
   - Para calcular el signo de g_j en $X = \alpha_j$ se calculan las condiciones de signo factibles de las derivadas de L y de g_j en $L = 0$. Por el Teorema 1.22 esto puede hacerse con complejidad

$$O(\ell^2 \operatorname{máx}\{\ell, d_X\} \log(\ell) \log^2(\operatorname{máx}\{\ell, d_X\})).$$

La complejidad total de este algoritmo es de $O\Big(t(\ell d_Y)^4 (\mathcal{H} + (\ell d_Y)^6 (\ell d_Y + \log(\mathcal{H}))^2)\Big)$. □

El análisis anterior nos permite, en particular, calcular de forma exacta la cantidad de raíces reales que un E-polinomio tiene en común con un polinomio en $\mathbb{Z}[X]$. Esto se enuncia en el siguiente resultado:

Corolario 3.10 *Dados polinomios $F \in \mathbb{Z}[X,Y]$, $h \in \mathbb{Z}[X]$, $\deg(h) > 0$, $L \in \mathbb{Z}[X]$ con grados acotados por d y altura acotada por H, y las codificaciones de Thom $\sigma_L(\alpha_1), \ldots, \sigma_L(\alpha_t)$ de las raíces reales $\alpha_1, \ldots, \alpha_t$ de L, se puede determinar $\#\{1 \leq j \leq t : F(\alpha_j, e^{h(\alpha_j)}) = 0\}$ con complejidad $O(d^3 \log^3(d))$. Más aún, los signos de $F(\alpha_j, e^{h(\alpha_j)})$, para $1 \leq j \leq t$, pueden calcularse con complejidad $O(8^d d^{3d+9} H^{2d})$.*

Demostración. Para la primera parte, observar que solo hay que aplicar los pasos 1 y 2 del algoritmo `E-SignDetermination`. Como $d_X, d_Y, \delta, \ell \leq d$, se tiene que la complejidad es de orden $O(d^3 \log^3(d))$.
Para la segunda parte, notar que

$$\mathcal{H} \leq \text{máx}\{(2d)!H^{2d}2^d, (2d)!H^{2d}(d+1)^d\} = (2d)!H^{2d}(d+1)^d \leq (2d)!H^{2d}(2d)^d.$$

Luego, $t(\ell d_Y)^4(\mathcal{H} + (\ell d_Y)^6(\ell d_Y + \log(\mathcal{H}))^2) = O\Big(d^9\big((2d)!H^{2d}2^d d^d + d^{12}(d^2 + \log(\mathcal{H}))^2\big)\Big)$. Usando que $(2d)! = O(4^d d^{2d})$ y que $\log(\mathcal{H}) = O(d^2 + dH)$, se obtiene que la complejidad total es de orden $O(8^d d^{3d+9} H^{2d})$. □

3.3. Cantidad de ceros de un E-polinomio

En esta sección utilizaremos el Algoritmo `E-SignDetermination` visto en la Sección 3.2 como una subrutina del Algoritmo `ZeroCounting`, descripto en la Subsección 2.2.3, para obtener un algoritmo que cuente los ceros de un E-polinomio sin llamadas a un oráculo.

El resultado más importante de esta sección se enuncia a continuación:

Teorema 3.11 *Sea $f(x) = F(x, e^{h(x)})$ un E-polinomio definido por $F \in \mathbb{Z}[X,Y]$ y $h \in \mathbb{Z}[X]$ con $\deg(F), \deg(h) \leq d$ y $H(F), H(h) \leq H$, y sea $[a,b]$ ($a, b \in \mathbb{Q}$) un intervalo cerrado.*

Supongamos que $\text{Res}_Y(F, \widetilde{F}) \neq 0$. Entonces existe un algoritmo que calcula la cantidad de ceros de f en $[a,b]$ con complejidad $(2dH)^{O(d^6)}$.

Demostración. Para probar este Teorema, adaptaremos el Algoritmo `ZeroCounting` introducido en la Subsección 2.2.3 para contar la cantidad de ceros de un E-polinomio sin llamadas a un oráculo. Para esto, es suficiente mostrar cómo se llevan a cabo los pasos 5 y 6 de este algoritmo y calcular su complejidad.

El paso 5 puede realizarse verificando la condición 3.2 para los números racionales a y b y computando los pasos 1 y 2 del Algoritmo `E-SignDetermination`aplicado al E-polinomio f al polinomio L definido en (2.8), en el paso 3 del algoritmo `ZeroCounting`. Como en este caso, si $\deg(h) = \delta$, $\delta_X = \delta - 1, \delta_Y = 1, d_X, d_Y, \delta \leq d$, se tiene que $\deg(L) \leq 10d^3$, donde L es el polinomio definido en (2.8), en el paso 3 del algoritmo `ZeroCounting`. Luego, por el Corolario 3.10, este paso puede realizarse con complejidad de orden $O(d^9 \log^3(d))$.

El paso 6 del algoritmo puede realizarse aplicando el Algoritmo `E-SignDetermination` a las funciones $f_{I_j,i}$ y sus derivadas sucesivas, para $0 \leq i \leq N$, donde $N \leq d_Y$. Las primeras están definidas salvo un signo por los polinomios R_i (introducidos en la Notación 2.24) y sus derivadas, por polinomios $S_{i\nu}$ tales que $\Big(R_i(x, e^{h(x)})\Big)^{(\nu)} = S_{i\nu}(x, e^{h(x)})$, para $0 \leq i \leq N$. Teniendo en cuenta que:

- Como $\deg_Y(\widetilde{F}) = \deg_Y(F)$, se tiene que $F_1 = \mathrm{cp}(F)^2\widetilde{F} - \mathrm{cp}(\widetilde{F})\mathrm{cp}(F)F$ y entonces, por lo visto en la demostración del Corolario 2.15,

$$\deg_X(F_1) \leq 4d-1, \quad \text{y} \quad \deg_Y(F_1) \leq d-1.$$

Además, por las Proposiciones 1.11 y 3.2, vale que $H(\mathrm{cp}(F)^2) \leq (d+1)H^2$ y $H(\mathrm{cp}(F)\mathrm{cp}(\widetilde{F})) \leq (d+1)H^2(d+d^3H)$, de donde se sigue que

$$H(F_1) \leq H(\mathrm{cp}(F)^2\widetilde{F}) + H(\mathrm{cp}(\widetilde{F})\mathrm{cp}(F)F) \leq$$

$$\leq 2d(d+1)H^3(d+d^3H)+(d+1)^2H^3(d+d^3H) \leq 4d(d+1)H^3(d+d^3H) \leq 8(d+1)d^4H^4.$$

Luego, por Proposición 1.38 aplicada a F y F_1, se tiene que $\deg_Y(R_i) \leq d$ y que $\deg_X(R_i) \leq (d-1)d + d(4d-1) = 5d^2 - 2d$, para todo $i = 0, \ldots, N$.

- Por la Proposición 1.38 y el Lema 1.41 aplicados a F y F_1 se tiene que:
  - $\deg(\tau_0) \leq d$, $\deg(\tau_1) \leq 4d-1$, $\deg(\rho_1) \leq d$, $H(\tau_0) \leq H(F)$, $H(\tau_1) \leq H(F_1)$ y $H(\rho_1) \leq H(F)$; y
  - para $2 \leq i \leq N$, $3 \leq j \leq N+1$, $\deg(\tau_i), \deg(\rho_j) \leq d(5d-1) \leq 5d^2-1$ y $H(R_i) \leq (2d-1)!(d+1)^{d-1}H^{d-1}(4d)^d(8(d+1)d^4H^4)^d = (2d-1)!H^{5d-1}(d+1)^{2d-1}2^{5d}d^{5d}$. Usando que

$$(d+1)^{2d-1}(2d-1)! \leq (d+1)^{2d-1}(2d)^{2d-1} = \big(\frac{d+1}{d}\big)^{2d-1}2^{2d-1}d^{4d-2} \leq e^2 2^{2d-1}d^{4d-2},$$

obtenemos que $H(R_i) \leq H^{5d-1}2^{7d+2}d^{9d-2}$.

Aplicando la Proposición 1.11, se tiene que

$$H(L) \leq d(4d-1)(5d^2)^{2N-2}H8(d+1)d^4H^4\prod_{i=2}^{N}H(\tau_i)\prod_{i=3}^{N+1}H(\rho_i) \leq$$

$$8\ 5^{2d-2}d^{4d+1}H^5(4d-1)(d+1)\big(H^{5d-1}2^{7d+2}d^{9d-2}\big)^{2d-2} \leq$$

$$\leq 5^{2d-2}(d+1)^2H^{(5d-1)(2d-2)+5}2^{(7d+2)(2d-2)+5}d^{(9d-2)(2d-2)+5+2(2d-2)} \leq$$

$$\leq 5^{2d-2}H^{10d^2-12d+7}2^{14d^2-10d+1}d^{18d^2-18d+7}.$$

- La cota para la multiplicidad de un cero de un E-polinomio dada en la Proposición 3.3 aplicada a cada R_i implica que necesitaremos las derivadas de orden

$$\nu \leq 2(5d^2-2d)d + d^2 = 10d^3 - 3d^2,$$

para todo $0 \leq i \leq N$. Por otro lado, las cotas del Lema 3.2 aplicadas a los polinomios R_i implican que, para $\nu \leq 10d^3 - 3d^2$,

$$\deg_X(S_{i\nu}) \leq \nu(\deg(h)-1) + \deg_X(R_i) \leq (10d^3-3d^2)(d-1)+5d^2-2d \leq 10d^4-5d^3 \quad \text{y}$$

$$H(S_{i\nu}) \leq H(R_i)\big((10d^3-3d^2)(d-1)+5d^2-2d+d^3H\big)^\nu \leq H(R_i)\big(10d^4+(H-5)d^3\big)^{10d^3-3d^2}.$$

Luego, la complejidad de aplicar el algoritmo `E-SignDetermination` a cada uno de estos polinomios, que son a lo sumo $d(10d^3 - 3d^2)$, es de orden

$$O\left(d^{19}\big(\mathcal{H} + d^{24}(d^4 + \log(\mathcal{H}))^2\big)\right),$$

donde $\mathcal{H}$ es del orden del máximo entre $(10d^3+d)!H(L)^d(2H)^{10d^3}$ y $(10d^4+5d^3)!H(L)^{10d^4-5d^3}$ $\times ((d+1)H^{5d-1}2^{7d+2}d^{9d-2}(10d^4+(H-5)d^3)^{10d^3-3d^2})^{10d^3}$, que resulta ser del orden de

$$(2dH)^{O(d^6)}.$$

Finalmente, para cada intervalo I_j, los signos $\mathrm{sg}(f_{I_j,i}, \alpha_j^+)$ y $\mathrm{sg}(f_{I_j,i}, \alpha_{j+1}^-)$ se obtienen a partir de los signos correspondientes de las funciones $R_i(x, e^{h(x)})$ siguiendo la Definición 2.9.

La complejidad total del algoritmo es de orden $(2dH)^{O(d^6)}$. □

El procedimiento anterior puede ser modificado para contar algorítmicamente la cantidad total de ceros en $\mathbb{R}$ de un E-polinomio. Para ello, consideraremos los signos de E-polinomios en $+\infty$ y $-\infty$.

Definición 3.12 *Sea $g(x) = G(x, e^{h(x)})$ un E-polinomio, donde $G(X,Y) = \sum_{j=0}^{d_Y} a_j(X)Y^j$ con $a_{d_Y} \neq 0$, y sea $j_0 = \min\{\ j : a_j \neq 0\ \}$. Definimos*

$$\mathrm{sg}(g, +\infty) = \begin{cases} \mathrm{sg}(\mathrm{cp}(a_{j_0})) & \text{si } \mathrm{cp}(h) < 0 \\ \mathrm{sg}(\mathrm{cp}(a_{d_Y})) & \text{si } \mathrm{cp}(h) > 0 \end{cases}$$

y

$$\mathrm{sg}(g, -\infty) = \begin{cases} \mathrm{sg}\big((-1)^{\deg(a_{j_0})}\mathrm{cp}(a_{j_0})\big) & \text{si } (-1)^{\deg(h)}\mathrm{cp}(h) < 0 \\ \mathrm{sg}\big((-1)^{\deg(a_{d_Y})}\mathrm{cp}(a_{d_Y})\big) & \text{si } (-1)^{\deg(h)}\mathrm{cp}(h) > 0. \end{cases}$$

De esta forma, dado que un E-polinomio tiene finitos ceros reales, se verifica que $\mathrm{sg}(g, +\infty)$ coincide con el signo que toma g cuando x es suficientemente grande, mientras que $\mathrm{sg}(g, -\infty)$ coincide con el signo que toma g cuando x es suficientemente chico.

Notación 3.13 *Dada una secuencia de E-polinomios $\mathbf{f} = (f_0, \ldots, f_N)$, notaremos $v(\mathbf{f}, +\infty)$ a la cantidad de cambios de signo de la secuencia $\big(\mathrm{sg}(f_0, +\infty), \ldots, \mathrm{sg}(f_N, +\infty)\big)$ y $v(\mathbf{f}, -\infty)$ a la cantidad de cambios de signo de la secuencia $\big(\mathrm{sg}(f_0, -\infty), \ldots, \mathrm{sg}(f_N, -\infty)\big)$.*

De esta forma se obtiene una generalización de la Proposición 2.13 a intervalos no acotados de la forma $(-\infty, M)$ o $(M, +\infty)$. Como consecuencia, vale la siguiente generalización del Teorema 2.14:

Teorema 3.14 *Sea $f(x) = F(x, e^{h(x)})$ un E-polinomio definido por $F \in \mathbb{Z}[X,Y]$ y $h \in \mathbb{Z}[X]$ con $\deg(F), \deg(h) \leq d$ y $H(F), H(h) \leq H$. Supongamos que $Res_Y(F, \widetilde{F}) \neq 0$. Sean además, ρ_i y τ_i los polinomios en $\mathbb{Z}[X]$ introducidos en la Notación 2.8 y $\alpha_1 < \alpha_2 < \ldots < \alpha_k$ todas las raíces reales de ρ_i, para $2 \leq i \leq N+1$ y τ_i, para $0 \leq i \leq N$. Teniendo en cuenta la*

Notación 2.8 y la Definición 2.9, sean $\mathbf{f}_{I_j}$*, para* $j = 0, \ldots, k$*, las secuencias de Sturm para* $f(x) = F(x, e^{h(x)})$ *en los intervalos* $I_j = (\alpha_j, \alpha_{j+1})$ *para* $j = 1, \ldots, k-1$*,* $I_0 = (-\infty, \alpha_1)$ *y* $I_k = (\alpha_k, +\infty)$ *respectivamente.*

La cantidad de ceros reales de f *coincide con*

$$\#\{1 \leq j \leq k \ / \ f(\alpha_j) = 0\} + v(\mathbf{f}_{I_0}, -\infty) - v(\mathbf{f}_{I_0}, \alpha_1^-) +$$

$$+ \sum_{j=1}^{k-1} v(\mathbf{f}_{I_j}, \alpha_j^+) - v(\mathbf{f}_{I_j}, \alpha_{j+1}^-) + v(\mathbf{f}_{I_k}, \alpha_k^+) - v(\mathbf{f}_{I_k}, +\infty).$$

En el Teorema 3.11 se pide como hipótesis que $\mathrm{Res}_Y(F, \widetilde{F}) \neq 0$. Según el siguiente resultado, esto puede ser evitado:

Proposición 3.15 *Sea* $f(x) = F(x, e^{h(x)})$ *un E-polinomio definido por* $F \in \mathbb{Z}[X, Y]$ *y* $h \in \mathbb{Z}[X]$ *con* $\deg(F), \deg(h) \leq d$ *y* $H(F), H(h) \leq H$*. Dado un intervalo de* $\mathbb{R}$*, existe un algoritmo que calcula la cantidad exacta de ceros de* f *en dicho intervalo con complejidad de orden* $\left(dH2^d\right)^{O(d^6)}$*.*

Demostración. Si $\mathrm{Res}_Y(F, \widetilde{F}) \neq 0$, el resultado sigue de los Teoremas 3.14 y 3.11.

Si $\mathrm{Res}_Y(F, \widetilde{F}) = 0$, en la demostración del Lema 2.7 se construye un polinomio $P \in \mathbb{Z}[X, Y]$ que verifica que $\mathrm{Res}_Y(P, \widetilde{P}) \neq 0$ y que $P(x, \varphi(x))$ tiene los mismos ceros reales que $f(x)$. Si $F = \mathrm{cont}(F)F_0$, P resulta ser $F/\gcd(F_0, \widetilde{F}_0)$ donde el divisor común mayor se calcula en $\mathbb{Z}[X, Y]$. Como P es un factor de F, por la Proposición 1.15, se tiene que $H(P) \leq 4^d(d+1)H$. Entonces, por el Teorema 3.11, la complejidad del algoritmo aplicado a P es del orden de $\left(dH2^d\right)^{O(d^6)}$. Notar que el cálculo de P a partir de F no modifica este orden de complejidad. □

3.4. Problema de decisión

En esta sección, utilizaremos los algoritmos descriptos anteriormente para resolver el problema de decisión introducido en la Sección 2.4 para fórmulas que involucran E-polinomios univariados. Daremos además, una breve aplicación a E-polinomios multivariados.

3.4.1. Problema de decisión para E-polinomios de una variable

En el Capítulo 2, al trabajar con funciones Pfaffianas arbitrarias de orden 1 en una variable, para determinar el signo que toman en un número real algebraico utilizamos un oráculo. Como vimos en la Sección 3.2, para E-polinomios, estos signos y consecuentemente los signos a la izquierda y a la derecha de un número algebraico (requeridos para calcular los indicadores de Tarski) pueden ser calculados explícitamente. Estimaremos a continuación la complejidad requerida para ello.

Proposición 3.16 *Sea $h \in \mathbb{Z}[X]$ un polinomio de grado $\delta > 0$ y, para $F \in \mathbb{Z}[X,Y]$ con $\deg_X(F) = d_X$ y $\deg_Y(F) = d_Y$, sea $f(x) = F(x, e^{h(x)})$. Sea $\alpha \in \mathbb{R}$ una raíz de un polinomio $L \in \mathbb{Z}[X]$ con $\deg(L) = \ell$ dada por su codificación de Thom como raíz de L. Entonces, se puede determinar $\mathrm{sg}(f, \alpha^+)$ y $\mathrm{sg}(f, \alpha^-)$ con complejidad*

$$O(\ell^4 d_Y^5 (d_X + \delta)(\mathcal{H} + (\ell d_Y)^6 (\ell d_Y + \log(\mathcal{H}))^2))$$

donde $\mathcal{H} = (\ell + d_X + d_Y(2d_X + \delta)(\delta - 1))! H(L)^{d_X + d_Y(2d_X+\delta)(\delta-1)} ((d_Y + 1)H(L)(d_Y(2d_X + \delta)(\delta - 1) + d_X + d_Y \delta^2 H(h))^{d_Y(2d_X+\delta)})^\ell$.

Demostración. Para calcular los signos $\mathrm{sg}(f, \alpha^+)$ y $\mathrm{sg}(f, \alpha^-)$ tendremos en cuenta la Observación 2.12. Luego, es suficiente calcular los signos de $f^{(i)}(\alpha)$ para $0 \leq i \leq \mathrm{mult}(\alpha, f)$. Por la Proposición 3.3, se tiene que $\mathrm{mult}(\alpha, f) \leq d_Y(2d_X + \delta)$, por lo que se necesita aplicar a lo sumo $d_Y(2d_X + \delta)$ veces el Algoritmo `E-SignDetermination` de la Sección 3.2 (con $t = 1$) a los polinomios F_i que definen a $f^{(i)}$. Teniendo en cuenta el Lema 3.2, para todo $i = 0, \ldots, \mathrm{mult}(\alpha, f)$, se verifica que:

- $\deg_X(F_i) \leq d_X + i(\delta - 1) \leq d_Y(2d_X + \delta)(\delta - 1) + d_X$,
- $\deg_Y(F_i) = d_Y$ y
- $H(F_i) \leq H(F) \prod_{j=0}^{i-1} (j(\delta - 1) + d_X + d_Y \delta^2 H(h)) \leq$

$$\leq H(F) \Big(d_Y(2d_X + \delta)(\delta - 1) + d_X + d_Y \delta^2 H(h) \Big)^{d_Y(2d_X+\delta)}.$$

Se obtiene entonces una complejidad de orden

$$O\Big(\ell^4 d_Y^5 (d_X + \delta)(\mathcal{H} + (\ell d_Y)^6 (\ell d_Y + \log(\mathcal{H}))^2)\Big),$$

donde $\mathcal{H} = (\ell + d_X + d_Y(2d_X + \delta)(\delta - 1))! H(L)^{d_X + d_Y(2d_X+\delta)(\delta-1)} ((d_Y + 1)H(P)(d_Y(2d_X + \delta)(\delta - 1) + d_X + d_Y \delta^2 H(h))^{d_Y(2d_X+\delta)})^\ell$. □

Como consecuencia, podemos estimar la complejidad de calcular indicadores de Tarski para E-polinomios:

Proposición 3.17 *Sean $f(x) = F(x, e^{h(x)})$ y $g(x) = G(x, e^{h(x)})$ los E-polinomios definidos por $F, G \in \mathbb{Z}[X,Y]$ y $h \in \mathbb{Z}[X]$ con $\deg(F), \deg(G), \deg(h) \leq d$ y $H(F), H(G), H(h) \leq H$. Sea (a, b) un intervalo, donde a, b son números algebraicos reales dados por su codificación de Thom, $-\infty$ o $+\infty$. Existe un algoritmo que calcula los indicadores de Tarski $\mathrm{TaQ}(f, g; a, b)$ con complejidad $(2dH)^{O(d^6)}$.*

Demostración. Consideremos primero el caso $a, b \in \mathbb{R}$.

El algoritmo está basado en el Algoritmo `Tarski-query` visto en la Sección 2.3 pero cada llamada al oráculo será reemplazada por un llamado al algoritmo `E-SignDetermination` de acuerdo a la Proposición 3.16. Para poder calcular la complejidad de este algoritmo, debemos acotar los grados y la altura de los polinomios involucrados:

- Por el Lema 2.23, se tiene que $\deg_X(\widetilde{F}G) \leq 3d-1$ y $\deg_Y(\widetilde{F}G) \leq 2d$ y, por lo tanto,

$$\deg_X(R_i) \leq 5d^2 - d \ \text{ y } \ \deg_Y(R_i) \leq d \ \text{ para } \ 0 \leq i \leq N$$

(ver Notación 2.24 y Proposición 1.38).

- Como $H(\widetilde{F}) \leq Hd(1+d^2H)$ y $\deg(\widetilde{F}) \leq 2d-1$ (ver Proposición 3.2), la Proposición 1.11 implica que

$$H(\widetilde{F}G) \leq 4d^3H^2(1+d^2H)(d+1)^2 \leq 32d^7H^3.$$

Luego, aplicando la Proposición 1.38, se tiene que

$$H(R_i) \leq (3d)!(32d^7H^3(3d))^d(H(d+1))^{2d} = (3d)!(d+1)^{2d}3^d2^{5d}d^{8d}H^{5d}.$$

Usando que $(3d)!(d+1)^{2d} \leq 3^{3d}d^{5d}$, obtenemos que $H(R_i) \leq 2^{5d}3^{4d}H^{5d}d^{13d}$.

- Como $\deg \rho_i$ y $\deg \tau_i$ son menores o iguales que $2d(d)+d(3d-1) = 5d^2-d$ y sus alturas son menores o iguales que $2^{5d}3^{4d}H^{5d}d^{13d}$ para $0 \leq i \leq N$ (ver Corolario 1.40), se tiene que

$$\deg(L) \leq (2d+3)(5d^2-d) \ \text{ y}$$

$$H(L) \leq (2^{5d}3^{4d}d^{13d}H^{5d})^{2d+3}(5d^2-d+1)^{2d+3} \leq (5\cdot 2^{5d}3^{4d}d^{13d+2}H^{5d})^{2d+3}$$

(ver Proposición 1.11).

A partir de estas cotas, se deduce que el cálculo de cada uno de los signos $\operatorname{sg}(R_i(x,e^{h(x)}),\alpha_j^+)$ o $\operatorname{sg}(R_i(x,e^{h(x)}),\alpha_j^-)$, por la Proposición 3.16 (con $\ell = \deg(L)$, $d_Y = d$ y $d_X = 5d^2-d$) puede efectuarse con complejidad

$$O\big(d^{12}d^5d^2(\mathcal{H}+d^{24}(d^4+\log(\mathcal{H}))^2)\big) \leq O\big(d^{19}(\mathcal{H}+d^{24}(d^4+\log(\mathcal{H}))^2)\big),$$

donde $\mathcal{H} \leq \big((2d+3)(5d^2-d)+5d^2-d+d(10d^2-d)(d-1)\big)!(5.\,2^{5d}3^{4d}d^{13d+2}H^{5d})^{(2d+3)(10d^4-5d^3)}\times$

$\big((d+1)(2^{5d}3^{4d}d^{13d+2}H^{5d})^{2d+3}(10d^4-5d^3+d^3H)^{10d^3-d^2}\big)^{(2d+3)(5d^2-d)} = O\big((2dH)^{O(d^6)}\big)$.

Notar que calcular todos estos signos y los de $f(\alpha_j)$ y $g(\alpha_j)$ no incrementa el orden de la complejidad.

En caso de que $\alpha = -\infty$ o $\beta = +\infty$, se debe adaptar el Algoritmo `Tarski-query` para usarlo en intervalos no acotados. Para ello, basta determinar los signos que la secuencia de E-polinomios toma en $-\infty$ o $+\infty$. Esto se hace teniendo en cuenta la Definición 3.12. □

Similarmente a lo hecho en la Sección 2.4, a partir de la complejidad del cálculo de indicadores de Tarski, obtenemos la complejidad para la determinación de todas las condiciones de signo factibles sobre una familia finita de E-polinomios.

Proposición 3.18 *Sean $g_1,\ldots,g_s$ los E-polinomios definidos por $g_i(x) = G_i(x,e^{h(x)})$ donde $G_i \in \mathbb{Z}[X,Y]$ $(1 \leq i \leq s)$ y $h \in \mathbb{Z}[X]$ son polinomios de grados acotados por d y alturas acotadas por H. Existe un algoritmo que calcula todas las condiciones de signo factibles para $g_1,\ldots,g_s$ con complejidad $(2dH)^{O(s^7d^6)}$.*

Demostración. Basta determinar las condiciones de signo factibles para $g_1, \ldots, g_s$ sobre los ceros de $f := \prod_{i=1}^{s} g_i$ y f' y decidir los signos de $g_1, \ldots, g_s$ en $-\infty$ y en $+\infty$.

Sea $F(X,Y) = \prod_{i=1}^{s} G_i(X,Y)$. Luego, $\deg(F) \leq sd$ y, por la Proposición 1.11, $H(F) \leq H^s(d+1)^{2s}$. Por el Lema 3.2, se obtiene que

$$\deg(\widetilde{F}) \leq \delta - 1 + \deg(F) \leq sd + d \text{ y}$$

$$H(\widetilde{F}) \leq H^s(d+1)^{2s} sd(1+d^2H) \leq 2H^{s+1}(d+1)^{2s} sd^3.$$

La cantidad de ceros de f y la cantidad de ceros de f' son, por el Corolario 2.15 aplicado a f y f', de orden $O(s^4d^4)$.

Por la Proposición 3.17, la complejidad de calcular cada indicador de Tarski del tipo $\mathrm{TaQ}(f, g_1^{\alpha_1} \ldots g_i^{\alpha_i})$ o del tipo $\mathrm{TaQ}(f', g_1^{\alpha_1} \ldots g_i^{\alpha_i})$ con $\alpha_j \in \{0,1,2\}$ es de orden $(2dH)^{O(s^7d^6)}$.

Notar que la cantidad de indicadores de Tarski necesarios, la resolución de los sistemas lineales y calcular los signos en $-\infty$ y en $+\infty$ no modifican el orden de la complejidad. □

Como consecuencia, podemos resolver algorítmicamente el problema de decisión para E-polinomios sin necesidad de un oráculo:

Teorema 3.19 *Sea Ψ una fórmula prenexa de longitud $|\Psi|$ en una variable cuantificada x que involucra E-polinomios $g_1, \ldots, g_s$ definidos por $g_i(x) = G_i(x, e^{h(x)})$ para $G_i \in \mathbb{Z}[X,Y]$ y $h \in \mathbb{Z}[X]$ polinomios con grados acotados por d y alturas acotadas por H. Existe un algoritmo que determina si Ψ es verdadera o falsa con complejidad $(2dH)^{O(s^7d^6)} + O(s^4d^4|\Psi|)$.*

3.4.2. Un problema de decisión para E-polinomios de varias variables

El método presentado en la sección anterior puede extenderse a E-polinomios de varias variables, es decir, a funciones Pfaffianas de la forma $f(x_1, \ldots, x_n) = F(x_1, \ldots, x_n, e^{h(x_1,\ldots,x_n)})$ donde $F \in \mathbb{Z}[X_1, \ldots, X_n, Y]$ y $h \in \mathbb{Z}[X_1, \ldots, X_n]$.

Primero, definiremos el problema de consistencia para estas funciones. Para $F_1, \ldots, F_s \in \mathbb{Z}[X_1, \ldots, X_n, Y]$ y $h \in \mathbb{Z}[X_1, \ldots, X_n]$, consideremos la fórmula

$$\exists \mathbf{x} : F_1(\mathbf{x}, e^{h(\mathbf{x})})\epsilon_1 0 \wedge \cdots \wedge F_s(\mathbf{x}, e^{h(\mathbf{x})})\epsilon_s 0$$

con $\epsilon_i \in \{<, >, =\}$ para $1 \leq i \leq s$, y $\mathbf{x} = (x_1, \ldots, x_n)$. Esta fórmula es equivalente a

$$\exists z \exists \mathbf{x} : F_1(\mathbf{x}, e^z)\epsilon_1 0 \wedge \cdots \wedge F_s(\mathbf{x}, e^z)\epsilon_s 0 \wedge z = h(\mathbf{x}). \tag{3.3}$$

Consideremos la fórmula polinomial

$$\exists \mathbf{x} : F_1(\mathbf{x}, y)\epsilon_1 0 \wedge \cdots \wedge F_s(\mathbf{x}, y)\epsilon_s 0 \wedge z = h(\mathbf{x}).$$

Por la eliminación de cuantificadores sobre $\mathbb{R}$, esta fórmula es equivalente a una fórmula libre de cuantificadores $\psi(z, y)$. Luego, la fórmula (3.3) es equivalente a

$$\exists z \psi(z, e^z)$$

y, aplicando el Teorema 3.19, podemos decidir si esta fórmula es falsa o verdadera.

Similarmente, podemos trabajar con el problema de decisión para cualquier fórmula prenexa con un solo bloque de cuantificadores existenciales, todas las variables cuantificadas y E-polinomios con h fijo, y obtener cotas de complejidad.

Proposición 3.20 *Sea Ψ una fórmula libre de cuantificadores en las variables $\mathbf{x} = (x_1, \ldots, x_n)$ definida por $g_i(\mathbf{x}) = G_i(\mathbf{x}, e^{h(\mathbf{x})})$ para $G_i \in \mathbb{Z}[X_1, \ldots, X_n, Y]$, para $1 \leq i \leq s$, y $h \in \mathbb{Z}[X_1, \ldots, X_n]$ polinomios con grados acotados por d y alturas acotadas por H. Existe un algoritmo simbólico que determina si la fórmula $\exists x_1 \ldots \exists x_n \Psi(x_1, \ldots, x_n)$ es verdadera o falsa con complejidad $(2dH)^{(sd)^{O(n)}}$.*

Demostración. El resultado se sigue directamente de las consideraciones previas aplicando las cotas de complejidad de [3, Theorem 14.22] para la eliminación de cuantificadores sobre $\mathbb{R}$ y el Teorema 3.19. □

3.5. Otros resultados sobre E-polinomios

En esta sección mostraremos cómo otros resultados válidos para polinomios pueden generalizarse a los E-polinomios. En la Subsección 3.5.1, daremos una cota superior para el tamaño de los ceros de un E-polinomio. En la Subsección 3.5.2, daremos la forma de construir un E-polinomio con suficientes ceros de modo de dar una respuesta negativa a una conjetura planteada en [19] sobre la cantidad de ceros de un E-polinomio. En la Subsección 3.5.3, introduciremos una codificación para los ceros de estas funciones inspirada en la codificación de Thom para ceros de polinomios y analizaremos la complejidad de calcularla. Finalmente, en la Subsección 3.5.4, daremos una generalización del Teorema de Budan-Fourier a E-polinomios.

3.5.1. Tamaño de los ceros

En esta sección encontraremos un intervalo acotado que contiene todos los ceros reales de un E-polinomio, cuyos extremos se calculan en función de los grados y alturas de los polinomios involucrados en su definición. Usando esta cota y aplicando sucesivamente nuestro algoritmo para contar ceros sería posible, mediante un método del tipo bisección iterada, por ejemplo, aproximar ceros de un E-polinomio.

Sea $f(x) = F(x, e^{h(x)})$ un E-polinomio definido por polinomios

$$F(X,Y) = \sum_{i=0}^{d_Y} a_i(X)\, Y^i \in \mathbb{Z}[X,Y] \qquad \text{y} \qquad h(X) = \sum_{k=0}^{\delta} h_k X^k \in \mathbb{Z}[X].$$

Supongamos que $a_{d_Y}(X) \neq 0$, $\deg_X(F) \leq d_X$, $\deg(h) = \delta \geq 1$, y sean $H, T \in \mathbb{Z}$ cotas superiores de las alturas de los polinomios F y h respectivamente.

Teorema 3.21 *Con las hipótesis y notaciones anteriores, si $\alpha \in \mathbb{R}$ es tal que $f(\alpha) = 0$, se tiene que*

$$|\alpha| < \text{máx}\left\{3H,\ 4T+1, (\frac{8d_X}{\delta}\ln(d_X))^{1/\delta}\right\}.$$

Para probar el resultado anterior, usaremos el siguiente lema con algunas cotas auxiliares:

Lema 3.22 *Sea* $g(X) = \sum_{i=0}^{n} c_i X^i \in \mathbb{Z}[X]$ *un polinomio de grado* $n \geq 1$ *y altura menor o igual que* $\Lambda \in \mathbb{N}$. *Entonces,*

(a) para todo $\alpha \in \mathbb{C}$ *con* $|\alpha| \geq 2$, *se tiene que* $|g(\alpha)| \leq 2\Lambda|\alpha|^n - 1$.

(b) si $k \geq 1$, *para todo* $\alpha \in \mathbb{C}$ *con* $|\alpha| \geq k\Lambda + 1$, *se tiene que* $|g(\alpha)| > (1 - \frac{1}{k})|\alpha|^n$.

Demostración.

(*a*) Si $|\alpha| \geq 2$, se tiene que

$$|g(\alpha)| \leq \sum_{i=0}^{n} |c_i||\alpha|^i \leq \Lambda|\alpha|^n + \Lambda \sum_{i=0}^{n-1} |\alpha|^i = \Lambda(|\alpha|^n + \frac{|\alpha|^n - 1}{|\alpha| - 1}) \leq$$

$$\leq \Lambda(|\alpha|^n + |\alpha|^n - 1) \leq 2\Lambda|\alpha|^n - 1.$$

(*b*) Para todo α con $|\alpha| \neq 1$, valen las siguientes desigualdades:

$$|g(\alpha)| \geq |c_n||\alpha|^n - \sum_{i=0}^{n-1} |c_i||\alpha|^i \geq |\alpha|^n - \Lambda \sum_{i=0}^{n-1} |\alpha|^i = |\alpha|^n - \Lambda(\frac{|\alpha|^n - 1}{|\alpha| - 1}).$$

Como $|\alpha| \geq k\Lambda + 1$, se sigue que $\dfrac{\Lambda}{|\alpha| - 1} \leq \dfrac{1}{k}$. Luego, las desigualdades anteriores implican que

$$|g(\alpha)| \geq |\alpha|^n - \frac{1}{k}(|\alpha|^n - 1) > (1 - \frac{1}{k})|\alpha|^n.$$

□

Ahora estamos en condiciones de probar el Teorema 3.21:

Demostración. Si $\deg_X(F) = 0$, por el Lema 3.4, si α es un cero de f entonces $h(\alpha) = 0$. Luego, por el Lema 1.16, se tiene que $|\alpha| < 1 + T$ y el teorema vale.

Si $\deg_X(F) \geq 1$, sin pérdida de generalidad, podemos asumir que $a_0(X) \not\equiv 0$, pues $e^{h(x)} \neq 0\ \forall x \in \mathbb{R}$.

Notemos $d_i := \deg(a_i)$, para todo $0 \leq i \leq d_Y$ tal que $a_i(X) \not\equiv 0$.

Sea $\alpha \in \mathbb{R}$ tal que $|\alpha| \geq \text{máx}\left\{3H,\ 4T + 1, (\dfrac{8d_X}{\delta} \ln(d_X))^{1/\delta}\right\}$.

Como $|\alpha| \geq 3H \geq 1 + 2H \geq 2$, si $d_i \geq 1$:

- Por el Lema 3.22 *b*) aplicado a a_i para $k = 2$, se tiene que $|a_i(\alpha)| > \frac{1}{2}|\alpha|^{d_i} \geq 1$.

- Por el Lema 3.22 *a*), aplicado a a_i se tiene que $|a_i(\alpha)| \leq 2H|\alpha|^{d_i} - 1 \leq 2H\,|\alpha|^{d_X} - 1$.

Observar que las cotas anteriores también son ciertas si $d_i = 0$ pues en ese caso

$$1 \leq |a_i(\alpha)| \leq H \leq 2H\,|\alpha|^{d_X} - 1.$$

Luego, aplicando el Lema 1.16 al polinomio $F(\alpha, Y) \in \mathbb{R}[Y]$, obtenemos que si $F(\alpha, \beta) = 0$,

$$|\beta| < 1 + \max_i \left\{ \frac{|a_i(\alpha)|}{|a_{d_Y}(\alpha)|} \right\} \leq 1 + \max_i \left\{ 2H\,|\alpha|^{d_X} - 1 \right\} = 2H|\alpha|^{d_X}.$$

Análogamente para el polinomio $F^*(Y) := Y^{d_Y} F(\alpha, 1/Y)$:

Si $F(\alpha, \beta) = 0$, $\beta \neq 0$ pues $a_0(\alpha) \neq 0$. Luego, $F^*(1/\beta) = 0$ y se tiene que $|\beta| > (2H|\alpha|^{d_X})^{-1}$.

Obtenemos entonces que, si $|\alpha| \geq 3H$, para toda raíz $\beta \in \mathbb{R}$ de $F(\alpha, Y)$ se satisfacen las desigualdades

$$(2H|\alpha|^{d_X})^{-1} < |\beta| < 2H|\alpha|^{d_X}. \tag{3.4}$$

Veamos que si $|\alpha|$ excede la cota del enunciado, entonces $\beta = e^{h(\alpha)}$ no verifica una de las desigualdades anteriores y, por lo tanto, α no puede ser un cero de f:

Aplicando el ítem (b) del Lema 3.22 al polinomio h con $k = 4$, se tiene que

$$|h(\alpha)| > \frac{3}{4}|\alpha|^{\delta}$$

para todo $\alpha \in \mathbb{R}$ tal que $|\alpha| \geq 4T + 1$. Luego,

$$e^{h(\alpha)} > e^{\frac{3}{4}|\alpha|^{\delta}} \quad \text{si } h(\alpha) > 0 \qquad \text{y} \qquad e^{h(\alpha)} < e^{-\frac{3}{4}|\alpha|^{\delta}} \quad \text{si } h(\alpha) < 0. \tag{3.5}$$

Por (3.4) y (3.5), alcanza con mostrar que $e^{\frac{3}{4}|\alpha|^{\delta}} \geq 2H|\alpha|^{d_X}$ o, equivalentemente, que

$$\frac{3}{4}|\alpha|^{\delta} \geq \ln(2H) + d_X \ln(|\alpha|). \tag{3.6}$$

Primero, notar que si $|\alpha| \geq 3H$, entonces

$$\frac{1}{4}|\alpha|^{\delta} \geq \frac{1}{4}|\alpha| \geq \frac{3}{4}H \geq \ln(2H) \tag{3.7}$$

(la última desigualdad vale para todo $H \in \mathbb{N}$).

Por otro lado, si $d_X \geq 2$, para $|\alpha| \geq \left(\frac{8d_X}{\delta} \ln(d_X)\right)^{1/\delta}$, se tiene que

$$\frac{1}{2}|\alpha|^{\delta} > d_X \ln(|\alpha|), \tag{3.8}$$

pues la función $m(t) = \frac{1}{2}t^{\delta} - d_X \ln(t)$ es estrictamente creciente en $(\left(\frac{2d_X}{\delta}\right)^{1/\delta}, +\infty)$ y

$$m\Big(\Big(\frac{8d_X}{\delta} \ln(d_X)\Big)^{1/\delta}\Big) = \frac{4d_X}{\delta} \ln(d_X) - \frac{d_X}{\delta} \ln\Big(\frac{8d_X}{\delta} \ln(d_X)\Big) = \frac{d_X}{\delta} \ln\Big(\frac{d_X^3 \delta}{8 \ln(d_X)}\Big) > 0$$

(notar que $d_X^3 \delta \geq d_X^3 > 8 \ln(d_X)$ para $d_X \geq 2$). Si $d_X = 1$ y $|\alpha| \geq 2$, se tiene que $\frac{1}{2}|\alpha| > \ln(|\alpha|)$; luego, la desigualdad (3.8) también vale en este caso.

Juntando las desigualdades (3.7) y (3.8), obtenemos la desigualdad deseada (3.6). $\square$

Ejemplo 3.23 *Siguiendo con la misma notación, los siguientes ejemplos muestran que la cota para el módulo de un cero de un E-polinomio debe depender de H, T y d_X.*

1. *Sea $f(x) = (x - H)e^x + x - H$.*

   Entonces $F(X,Y) = (X - H)Y + X - H$ y $h(X) = X$. Un cero de f es $\alpha = H$.

2. *Sea $f(x) = e^{x-T} - 1$.*

   Entonces $F(X,Y) = Y - 1$ y $h(X) = X - T$. Un cero de f es $\alpha = T$.

3. *Sea $f(x) = x^{d_X} e^{-x} - 1$ con $d_X \geq 3$.*

   Entonces, $F(X,Y) = X^{d_X}Y - 1$ y $h(X) = -X$. Como $f(d_X \ln(d_X)) = \ln^{d_X}(d_X) - 1 > 0$ para $d_X \geq 3$, y $\lim_{x\to+\infty} f(x) = -1 < 0$, se deduce que f tiene un cero $\alpha > d_X \ln(d_X)$.

3.5.2. Un contraejemplo

En [19], se plantea el problema de determinar una cota ajustada para la cantidad de ceros de un E-polinomio en una variable de la forma $f(x) = F(x, e^x)$. El autor conjetura que, para todo E-polinomio definido por $F \in \mathbb{Z}[X,Y]$ con $\deg_X(F) = n$ y $\deg_Y(F) = m$, una cota superior para esta cantidad es $n + m$.

En esta subsección mostraremos que, dados $n, m \in \mathbb{N}$, existe un polinomio $F \in \mathbb{Z}[X,Y]$ no nulo, con $n = \deg_X(F)$ y $m = \deg_Y(F)$, tal que el E-polinomio $f(x) = F(x, e^x)$ tiene al menos $N = (n+1)(m+1) - 1$ ceros distintos.

Comenzaremos construyendo un polinomio $F \in \mathbb{R}[X,Y]$ que cumpla lo anterior y que además verifique que los N ceros distintos del E-polinomio que define sean todos simples.

Dados $P_1 = (x_1, y_1), \ldots, P_N = (x_N, y_N) \in \mathbb{R}^2$ todos distintos entre sí, a determinar, buscamos $a = \big(a_{kl}\big)_{0\leq k\leq n,\ 0\leq l\leq m}$, $a_{kl} \in \mathbb{R}$, para que el polinomio $F(X,Y) = \sum_{0\leq k\leq n,\ 0\leq l\leq m} a_{kl} X^k Y^l$ sea no nulo y tenga a los puntos $P_1, \ldots, P_N$ como ceros. Es decir que, si $\hat{F}$ es el polinomio con coeficientes en $\{0,1\}$ en $(n+1)(m+1)+2$ variables definido por $\hat{F}(a, X, Y) := F(X,Y)$, buscamos una solución no trivial del sistema lineal en las variables a

$$S_0 : \begin{cases} \hat{F}(a, x_1, y_1) = 0 \\ \quad \vdots \\ \hat{F}(a, x_N, y_N) = 0. \end{cases}$$

Además, para cada $i = 1, \ldots, N$, definimos el sistema lineal

$$S_i := S_0 \cup \left\{ \frac{\partial \hat{F}}{\partial X}(a, x_i, y_i) + y_i \frac{\partial \hat{F}}{\partial Y}(a, x_i, y_i) = 0 \right\}.$$

Observar que, como $\frac{\partial \hat{F}}{\partial X}(a, X, Y) = \frac{\partial F}{\partial X}(X,Y)$ y $\frac{\partial \hat{F}}{\partial Y}(a, X, Y) = \frac{\partial F}{\partial Y}(X,Y)$, P_i resulta ser un cero simple de F si y sólo si S_i tiene solución única (la trivial).

Consideremos las coordenadas $X_1, Y_1, \ldots, X_N, Y_N$ de los puntos $P_1, \ldots, P_N$ como indeterminadas y veamos cuál es la matriz del sistema S_i.

Teniendo en cuenta que si $k = 1, \dots, n$ y $l = 0, \dots, m$,

$$\Big(\frac{\partial}{\partial X} + Y\frac{\partial}{\partial Y}\Big)(X^k Y^l) = kX^{k-1}Y^l + lX^kY^l = X^{k-1}Y^l(k + lX),$$

se tiene que la matriz del sistema S_i es

$$A_i = \begin{pmatrix} 1 & Y_1 & \dots & Y_1^m & X_1 & X_1Y_1 & \dots & X_1Y_1^m & \dots & X_1^n & \dots & X_1^nY_1^m \\ 1 & Y_2 & \dots & Y_2^m & X_2 & X_2Y_2 & \dots & X_2Y_2^m & \dots & X_2^n & \dots & X_2^nY_2^m \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ 1 & Y_N & \dots & Y_N^m & X_N & X_NY_N & \dots & X_NY_N^m & \dots & X_N^n & \dots & X_N^nY_N^m \\ 0 & Y_i & \dots & mY_i^m & 1 & Y_i(1+X_i) & \dots & Y_i^m(1+mX_i) & \dots & nX_i^{n-1} & \dots & X_i^{n-1}Y_i^m(n+mX_i) \end{pmatrix}.$$

Veamos que la matriz A_i es inversible. Para ello, basta considerar $i = 1$ (por simetría). Si evaluamos la matriz A_1 en $X_1 = Y_1 = 0$ y $X_i = Y_i^{m+1}$ para $i = 2, \dots, N$, obtenemos la matriz de tamaño $(N+1) \times (N+1)$:

$$\begin{pmatrix} 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ 1 & Y_2 & \dots & Y_2^m & Y_2^{m+1} & Y_2^{m+2} & \dots & Y_2^{2m} & \dots & Y_2^{n(m+1)} & \dots & Y_2^N \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ 1 & Y_N & \dots & Y_N^m & Y_N^{m+1} & Y_N^{m+2} & \dots & Y_N^{2m} & \dots & Y_N^{n(m+1)} & \dots & Y_N^N \\ 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 & \dots & 0 & 0 \ \dots & 0 \end{pmatrix},$$

cuyo determinante es

$$D_{m+1}(Y_2, \dots, Y_N) := \det \begin{pmatrix} Y_2 & \dots & Y_2^m & Y_2^{m+2} & \dots & \dots & Y_2^N \\ \vdots & & \vdots & \vdots & & & \vdots \\ \vdots & & \vdots & \vdots & & & \vdots \\ Y_N & \dots & Y_N^m & Y_N^{m+2} & \dots & \dots & Y_N^N \end{pmatrix}.$$

Notar que agregando una nueva variable Z, por ser un determinante de una matriz de Vandermonde,

$$\det \begin{pmatrix} Z & Z^2 & \dots & Z^m & Z^{m+1} & Z^{m+2} & \dots & \dots & Z^N \\ Y_2 & Y_2^2 & \dots & Y_2^m & Y_2^{m+1} & Y_2^{m+2} & \dots & \dots & Y_2^N \\ \vdots & & & \vdots & \vdots & \vdots & & & \vdots \\ \vdots & & & \vdots & \vdots & \vdots & & & \vdots \\ Y_N & Y_N^2 & \dots & Y_N^m & Y_N^{m+1} & Y_N^{m+2} & \dots & \dots & Y_N^N \end{pmatrix} \neq 0.$$

Si lo desarrollamos por la columna $m+1$ y evaluamos $Z = Y_1$, resulta ser igual a

$$\sum_{i=1}^{N} \pm Y_i^{m+1} D_{m+1}(Y_1, \dots, \hat{Y_i}, \dots, Y_N),$$

donde $(Y_1, \ldots, \hat{Y_i}, \ldots, Y_N) = (Y_1, \ldots, Y_{i-1}, Y_{i+1}, \ldots, Y_N)$. Luego, se tiene que alguno de estos sumandos es no nulo, por lo que $D_{m+1}(Y_2, \ldots, Y_N) \neq 0$. De esta forma, obtenemos que el polinomio

$$H_1(X_1, \ldots, X_N, Y_1, \ldots, Y_N) := \det(A_1) \in \mathbb{Z}[X_1, \ldots, X_N, Y_1, \ldots, Y_N]$$

es no nulo. Análogamente (por simetría), obtenemos que los polinomios

$$H_i(X_1, \ldots, X_N, Y_1, \ldots, Y_N) := \det(A_i) \in \mathbb{Z}[X_1, \ldots, X_N, Y_1, \ldots, Y_N]$$

resultan ser no nulos, para $i = 2, \ldots, N$.

Si elegimos $x_1, \ldots, x_N$ números reales algebraicos y linealmente independientes sobre $\mathbb{Q}$ tales que $\prod_{i=1}^{N} H_i(x_1, \ldots, x_N, Y_1, \ldots, Y_N)$ es no nulo, obtenemos que

$$\prod_{i=1}^{N} H_i(x_1, \ldots, x_N, e^{x_1}, \ldots, e^{x_N}) \neq 0,$$

pues por el Teorema de Lindermann-Weierstrass, N es el grado de trascendencia del conjunto $\{e^{x_1}, \ldots, e^{x_N}\}$.

Obtenemos entonces que una solución del sistema S_0 para estos $x_1 \ldots, x_N$ y $y_i = e^{x_i}$, para $i = 1, \ldots, N$, permite construir un polinomio $F \in \mathbb{R}[X, Y]$ tal que el E-polimomio $f(x) = F(x, e^x)$ verifica que $f(x_i) = 0$ y $f'(x_i) \neq 0$, para $i = 1, \ldots, N$.

La idea será ahora aproximar este polinomio $F \in \mathbb{R}[X, Y]$ por otro polinomio G con coeficientes en $\mathbb{Q}$, de forma tal que el E-polinomio definido por G tenga N ceros distintos. Para ello, supongamos que $x_1 < \ldots < x_N$ y procedamos de la siguiente manera.

Para todo $i = 1, \ldots, N$, consideremos intervalos disjuntos dos a dos $I_i = [\alpha_i, \beta_i]$ tal que $\alpha_i < x_i < \beta_i$ y f' no se anula en I_i. En particular, por la continuidad de f, se tiene que $f(\alpha_i) f(\beta_i) < 0$. Sean

$$I = [\alpha_1, \beta_N],\ \varepsilon = \min_{1 \leq i \leq N} \{|f(\alpha_i)|, |f(\beta_i)|\} > 0 \text{ y } M = \max\{1, |\alpha_1|, |\beta_N|\}.$$

Para todo $0 \leq i \leq n$, $0 \leq j \leq m$, existe $b_{ij} \in \mathbb{Q}$ tal que $|b_{ij} - a_{ij}| < \frac{\varepsilon}{2M^n e^{mM}(n+1)(m+1)}$. Sea

$$G(X, Y) := \sum_{0 \leq i \leq n,\ 0 \leq j \leq m} b_{ij} X^i Y^j \in \mathbb{Q}[X, Y].$$

Este polinomio verifica que para todo $x \in I$:

$$|G(x, e^x) - F(x, e^x)| \leq \sum_{0 \leq i \leq n,\ 0 \leq j \leq m} |b_{ij} - a_{ij}| |x|^i e^{jx} < \sum_{0 \leq i \leq n,\ 0 \leq j \leq m} |a_{ij} - b_{ij}| M^n e^{mM} < \frac{\varepsilon}{2}.$$

En particular, para todo $x \in \{\alpha_1, \ldots, \alpha_N, \beta_1, \ldots, \beta_N\}$, se tiene que

$$|G(x, e^x) - F(x, e^x)| < \frac{\varepsilon}{2} \leq \frac{1}{2} |F(x, e^x)|,$$

de donde se deduce que $\operatorname{sg}(G(x, e^x)) = \operatorname{sg}(F(x, e^x))$ para todo $x \in \{\alpha_1, \ldots, \alpha_N, \beta_1, \ldots, \beta_N\}$, y por lo tanto, $G(\alpha_i, e^{\alpha_i}) G(\beta_i, e^{\beta_i}) < 0$. Luego, $G(x, e^x)$ tiene un cero en cada intervalo I_i, para $i = 1, \ldots, N$.

3.5.3. Codificación de Thom

En esta subsección, definiremos una noción de codificación de Thom de un cero de un E-polinomio en una variable. Para ello, recordaremos las nociones de pseudo-grado y pseudo-derivada de un E-polinomio univariado introducidas en [16]. Utilizando el algoritmo de la Proposición 3.18, mostraremos que existe un algoritmo que calcula las codificaciones de Thom de los ceros de un E-polinomio definido por un polinomio con coeficientes enteros y estimaremos su complejidad.

Recordemos la notación introducida en (3.1): dado el E-polinomio $f(x) = F(x, e^{h(x)})$, con $\deg_Y(F) > 0$ y $\deg(h) > 0$, notamos

$$\widetilde{F}(X,Y) := \frac{\partial F}{\partial X}(X,Y) + h'(X)Y\frac{\partial F}{\partial Y}(X,Y),$$

de forma tal que $f'(x) = \widetilde{F}(x, e^{h(x)})$.

Definición 3.24 *Sea $f(x) = F(x, e^{h(x)})$ con $F \neq 0$, se define el pseudo-grado de f como*

$$\operatorname{pdeg}(f) = \begin{cases} \big(\deg_Y(F), \deg(F(X,0)\big) & \text{si } F(X,0) \neq 0 \\ (\deg_Y(F), 0) & \text{si } F(X,0) = 0. \end{cases}$$

Se define también la pseudo-derivada de f como

$$\operatorname{pder}(f) = \begin{cases} f'(x) & \text{si } \deg(F(X,0)) > 0 \\ f'(x)e^{-kh(x)} & \text{si } F(X,0) \in \mathbb{R},\ f'(x) \neq 0,\ Y^k \mid \widetilde{F}(X,Y) \text{ y } Y^{k+1} \nmid \widetilde{F}(X,Y) \\ 0 & \text{si } f'(x) = 0. \end{cases}$$

Será útil considerar las siguientes observaciones:

Observación 3.25 *Notar que pder(f) resulta ser un E-polinomio definido por el polinomio $\widetilde{F}$ en el primer caso, mientras que resulta ser el E-polinomio definido por el polinomio*

$$Q_f(X,Y) = \frac{\widetilde{F}(X,Y)}{Y^k}$$

en el segundo caso. Observar que si $k \geq 1$, $\deg_Y(Q_f) = \deg_Y(\widetilde{F}) - k = d_Y - k$ y $\deg_X(Q_f) = \deg_X(\widetilde{F}) \leq d_X + \delta - 1$.

Observación 3.26 *Por definición de la pseudo-derivada, para todo E-polinomio f se tiene que*

$$\operatorname{sg}(f'(x)) = \operatorname{sg}(pder(f)(x))\ \forall x \in \mathbb{R}.$$

La relación entre los pseudo-grados de f y el de $\operatorname{pder}(f)$ es la siguiente:

Lema 3.27 *Dado el E-polinomio $f(x) = F(x, e^{h(x)})$. Si $f \notin \mathbb{R}$, se tiene que*

$$\operatorname{pdeg}(\operatorname{pder}(f)) <_{lex} \operatorname{pdeg}(f),$$

donde $<_{lex}$ es el orden lexicográfico.

Demostración. Si $f \notin \mathbb{R}$, hay dos opciones:

- Si $\deg(F(X,0)) > 0$, como $\operatorname{pder}(f) = f'$, $\operatorname{pdeg}(\operatorname{pder}(f)) = (\deg_Y(\widetilde{F}), \deg(\widetilde{F}(X,0)))$. Observando que $\deg_Y(\widetilde{F}) = d_Y$ y $\widetilde{F}(X,0) = (F(X,0))'$, obtenemos

$$\operatorname{pdeg}(\operatorname{pder}(f)) <_{\text{lex}} (d_Y, \deg(F(X,0))) = \operatorname{pdeg}(f).$$

- Si $\deg(F(X,0)) = 0$ y $f' \neq 0$, la primera coordenada de $\operatorname{pdeg}(\operatorname{pder}(f))$ es $\deg_Y(Q_f) = d_Y - k < d_Y$, que resulta ser la primera coordenada de $\operatorname{pdeg}(f)$.

□

Si notamos $\operatorname{pder}^{(i)}$ al operador que aplica i veces el operador pder, como consecuencia de la proposición anterior, se obtiene que $\{\operatorname{pder}^{(i)}(f)\}_{i\in\mathbb{N}}$ es una familia finita. Notaremos

$$\operatorname{PDer}(f) := \{\operatorname{pder}^{(i)}(f) : 0 \leq i \leq D\},$$

donde $D = \min\{i \ / \ \operatorname{pder}^{(i+1)}(f) = 0\}$. Veamos la relación que hay entre la multiplicidad de un cero de f con la multiplicidad de un cero de $\operatorname{pder}(f)$.

Lema 3.28 *Sean f un E-polinomio, $c \in \mathbb{R}$ y $\mu \geq 1$. Entonces*

$$\operatorname{mult}(c,f) = \mu \text{ si y sólo si } \operatorname{mult}(c, \operatorname{pder}(f)) = \mu - 1 \quad y \ f(c) = 0.$$

Demostración. Si $\operatorname{pder}(f) = f'$, el resultado vale claramente. Si no, con la misma notación que la Definición 3.24, $\operatorname{pder}(f)(x) = f'(x)e^{-kh(x)}$. Veamos que en este caso vale el si y sólo si.

⇒) Claramente $f(c) = 0$. Por otro lado, como f es una función analítica, existe una función analítica g tal que $f(x) = (x-c)^\mu g(x)$ con $g(c) \neq 0$. Luego,

$$e^{kh(x)}\operatorname{pder}(f)(x) = (x-c)^{\mu-1} \underbrace{(\mu g(x) + (x-c)g'(x))}_{s(x)},$$

donde $s(c) = \mu g(c) \neq 0$. De aquí se deduce que $\operatorname{mult}(c, \operatorname{pder}(f)) = \mu - 1 \geq 0$.

⇐) Si $\operatorname{mult}(c, \operatorname{pder}(f)) = \mu - 1$, como f es una función analítica, existe una función analítica g tal que

$$f'(x)e^{-kh(x)} = \operatorname{pder}(f)(x) = (x-c)^{\mu-1}g(x),$$

con $g(c) \neq 0$. De aquí se deduce que c tiene multiplicidad $\mu - 1$ como cero de f' y, como $f(c) = 0$, c tiene multiplicidad μ como cero de f.

□

En vista de querer codificar ceros de E-polinomios, necesitaremos el siguiente resultado que generaliza la Proposición 1.23 (válida para polinomios) a los E-polinomios.

Proposición 3.29 *Sea $f_1, \ldots, f_s$ una familia de E-polinomios cerrada bajo la pseudo-derivación. Sea $\varepsilon : \{1, \ldots, s\} \to \{-1, 0, 1\}$. Luego $\{x \in \mathbb{R} \ / \ \operatorname{sg}(f_i(x)) = \varepsilon(i)$ para $1 \leq i \leq s\}$ es vacío o un punto o un intervalo abierto.*

Demostración. Lo probaremos por inducción en s. Si $s = 1$, $f_1 = 0$ y no hay nada que probar. Supongamos que lo probamos para s y veamos que vale para $s + 1$.

Sea $\{f_1, \ldots, f_s, f_{s+1}\}$ una familia de E-polinomios cerrada bajo la pseudo-derivación. Podemos suponer que f_{s+1} tiene el máximo pseudo-grado. Luego, por el Lema 3.27, se tiene que $\{f_1, \ldots, f_s\}$ es cerrada para la pseudo-derivación.

Por hipótesis inductiva, $A = \{x \in \mathbb{R} \ / \ \mathrm{sg}(f_i(x)) = \varepsilon(i) \text{ para } 1 \leq i \leq s\}$ es, o bien vacío, o bien un punto, o bien un intervalo abierto. En los dos primeros casos, el resultado vale claramente. Si A es un intervalo abierto, como $\mathrm{pder}(f_{s+1}) \in \{f_1, \ldots, f_s\}$, entonces $\mathrm{pder}(f_{s+1})$ tiene signo constante en A. Si $\mathrm{pder}(f_{s+1}) = 0$ se tiene que f_{s+1} es una función constante, de donde se sigue el resultado. Si $\mathrm{pder}(f_{s+1}) \neq 0$, como $\mathrm{sg}(\mathrm{pder}(f_{s+1})) = \mathrm{sg}(f'_{s+1})$, resulta que f_{s+1} es una función estrictamente monótona en A. De la continuidad de la función f_{s+1} se sigue el resultado. □

Corolario 3.30 *Dado un E-polinomio f, los ceros de f están unívocamente determinados por las condiciones de signo factibles de* $\mathrm{PDer}(f)$ *sobre* $\{x \in \mathbb{R} \ / \ f(x) = 0\}$.

Esto nos permite definir la codificación de Thom para un cero de un E-polinomio:

Definición 3.31 *Sean $f(x) = F(x, e^{h(x)})$ un E-polinomio, $D = \mathrm{mín}\{i \ / \ \mathrm{pder}^{(i+1)}(f) = 0\}$ y $\xi \in \mathbb{R}$. Sea $\varepsilon : \{0, \ldots, D\} \to \{-1, 0, 1\}$ con $\varepsilon(0) = 0$. Decimos que (f, ε) es la* codificación de Thom *como cero de f de ξ si $\{ \ x \in \mathbb{R} \ / \ \mathrm{sg}(\mathrm{pder}^{(i)}(f)(x)) = \varepsilon(i)$ para $0 \leq i \leq D\} = \{\xi\}$.*

Un resultado análogo a la Proposición 1.25 se aplica en nuestro contexto y nos permite utilizar codificaciones de Thom para ordenar todos los ceros reales de un E-polinomio dado:

Observación 3.32 *Sea f un E-polinomio y sean (f, ε_1) y (f, ε_2) con $\varepsilon_j : \{0, \ldots, D\} \to \{-1, 0, 1\}$ para $j = 1, 2$ las codificaciones de Thom de dos ceros reales diferentes ξ_1 y ξ_2 de f. Sea $k = \mathrm{máx}\{0 \leq i \leq D \ / \ \varepsilon_1(i) \neq \varepsilon_2(i)\}$ (luego $\varepsilon_1(k+1) = \varepsilon_2(k+1) \neq 0$).*

- *Si $\varepsilon_1(k+1) = \varepsilon_1(k+1) = 1$ entonces: $\xi_1 > \xi_2$ si y sólo si $\varepsilon_1(k) > \varepsilon_2(k)$.*
- *Si $\varepsilon_1(k+1) = \varepsilon_1(k+1) = -1$ entonces $\xi_1 > \xi_2$ si y sólo si $\varepsilon_1(k) < \varepsilon_2(k)$.*

El siguiente ejemplo sencillo muestra las codificaciones de Thom de los ceros de un E-polinomio:

Ejemplo 3.33 *Sea $f(x) = (6x-1)e^{2x} - (8x+1)e^x - 1$. Este E-polinomio está asociado a la función $\varphi(x) = e^x$ y está definido por el polinomio $F(X, Y) = (6X-1)Y^2 - (8X+1)Y - 1$.*

Como $F(X, 0) = -1$ y $\widetilde{F}(X, Y) = (12X + 4)Y^2 - (8X + 9)Y$, se tiene que

$$\mathrm{pder}^{(1)}(f)(x) = f'(x)e^{-x} = (12x + 4)e^x - (8x + 9),$$

que resulta ser un E-polinomio definido por el polinomio $F_1(X, Y) = (12X+4)Y - (8X+9)$.

Como $\deg(F_1(X, 0)) = 1$, se tiene que

$$\mathrm{pder}^{(2)}(f)(x) = (\mathrm{pder}^{(1)}(f))'(x) = (12x + 16)e^x - 8,$$

que resulta ser el E-polinomio definido por el polinomio $F_2(X,Y) = (12X+16)Y - 8$.

Como $F_2(X,0) = -8$ *y* $\widetilde{F}_2(X,Y) = (12X+28)Y$, *se tiene que*

$$\mathrm{pder}^{(3)}(f)(x) = (\mathrm{pder}^{(2)}(f))'(x)e^{-x} = 12x + 28,$$

que resulta ser el E-polinomio definido por el polinomio $F_3(X,Y) = (12X+28)$.

Como $\deg(F_3(X,0)) = 1$, *se tiene que*

$$\mathrm{pder}^{(4)}(f)(x) = (\mathrm{pder}^{(3)}(f))'(x) = 12,$$

que resulta ser el E-polinomio definido por el polinomio $F_4(X,Y) = 12$.

Como $(\mathrm{pder}^{(4)}(f))'(x) = 0$, *se tiene que* $\mathrm{pder}^{(5)}(f)(x) = 0$.

De esta forma se obtiene que la secuencia de pseudo-derivadas de f *es* $(\mathrm{pder}^{(i)}(f))_{1\leq i\leq 4}$, *donde*

$$\begin{aligned}
\mathrm{pder}^{(1)}(f)(x) &= (12x+4)e^x - (8x+9),\\
\mathrm{pder}^{(2)}(f)(x) &= (12x+16)e^x - 8,\\
\mathrm{pder}^{(3)}(f)(x) &= 12x + 28,\\
\mathrm{pder}^{(4)}(f)(x) &= 12.
\end{aligned}$$

Se puede ver que el E-polinomio f *tiene* 3 *ceros,* $\xi_1, \xi_2, \xi_3 \in \mathbb{R}$. *Las codificaciones de Thom de* ξ_1, ξ_2, ξ_3 *como ceros de* f *están dadas por*

$$\varepsilon_1 = (0,-1,-1,-1,1),\ \varepsilon_2 = (0,-1,-1,1,1),\ \varepsilon_3 = (0,1,1,1,1).$$

Comparándolas como en la Observación 3.32, se sigue que $\xi_1 < \xi_2$ *(pues* $\varepsilon_1(4) = \varepsilon_2(4) = 1$ *y* $\varepsilon_1(3) < \varepsilon_2(3)$*) y* $\xi_2 < \xi_3$ *(pues* $\varepsilon_2(i) = \varepsilon_3(i) = 1$ *para* $i = 4, 3$ *y* $\varepsilon_2(2) < \varepsilon_3(2)$*).*

Desafortunadamente, no se puede asegurar que la suma de dos ceros de E-polinomios sea un cero de un E-polinomio, con lo cual a partir de las codificaciones de Thom de dos ceros de E-polinomios no es posible obtener una codificación para su suma. Por ejemplo, la pregunta planteada en [19] sobre si el conjunto

$$L = \{x \in \mathbb{R} \ / \ F(x, e^x) = 0, F \in \mathbb{Q}[X,Y]\}$$

es cerrado bajo la suma tiene una respuesta negativa, como mostramos a continuación:

Supongamos que L es cerrado bajo la suma. Como $\ln(2) \in L$ (pues es un cero de $F(x,e^x)$ para $F(X,Y) = Y - 2$), se tiene que $\ln(2) + 1 \in L$. Luego, existe un polinomio no nulo $F \in \mathbb{Q}[X,Y]$ tal que $F(\ln(2)+1, 2e) = 0$ y, por lo tanto, e es algebraico sobre $\mathbb{Q}(\ln(2))$. Similarmente, $\ln(2) + \sqrt{2} \in L$ y, luego $e^{\sqrt{2}}$ es algebraico sobre $\mathbb{Q}(\ln(2) + \sqrt{2})$. Como consecuencia, el grado de trascendencia de $\mathbb{Q}(\sqrt{2}, \ln(2), e, e^{\sqrt{2}})$ sobre $\mathbb{Q}$ es 1, contradiciendo el hecho de que el conjunto $\{e, e^{\sqrt{2}}\}$ es algebraicamente independiente sobre $\mathbb{Q}$ por el Teorema de Lindemann-Weierstrass.

Nuestros resultados previos de la Sección 3.4 sobre la determinación algorítmica de las condiciones de signo factibles de una familia finita de E-polinomios nos permiten calcular algorítmicamente codificaciones de Thom para ceros de E-polinomios. Para estimar la complejidad de este cálculo, analizaremos en primer lugar los pseudo-grados de las pseudo-derivadas sucesivas de un E-polinomio y los grados de los polinomios que las definen.

Observación 3.34 *Sea $g(x) = G(x, e^{h(x)})$ un E-polinomio definido por $G \in \mathbb{Z}[X, Y]$ de grado total d_0 y $h \in \mathbb{Z}[X]$ con $\deg(h) = \delta$. Entonces, teniendo en cuenta la Observación 3.25, la pseudo-derivada $\operatorname{pder}(g)$ está definida por un polinomio con grado total acotado por $d_0+\delta-1$ y, además, si $\operatorname{pdeg}(g) = (m_0, n_0)$, se tiene que:*

- *Si $n_0 \neq 0$, $\deg(G(X, 0)) > 0$ y $\operatorname{pder}(g) = g'(x)$. Luego,*

$$\operatorname{pdeg}(\operatorname{pder}(g)) = (\deg_Y(G), \deg(\frac{\partial G}{\partial X}(X, 0))).$$

- *Si $n_0 = 0$, $G(X, 0)$ es constante y* $\operatorname{pder}(g) = \begin{cases} Q_g(x, e^{h(x)}) & si\ m_0 > 0 \\ 0 & si\ m_0 = 0. \end{cases}$.

  Luego, si $m_0 > 0$,

$$\operatorname{pdeg}(\operatorname{pder}(g)) = (\deg_Y(Q_g), \deg(Q_g(X, 0))).$$

De las cotas de grado vistas en la Observación 3.25, deducimos que

$$\operatorname{pdeg}(\operatorname{pder}(g)) = \begin{cases} (m_0, n_0 - 1) & \text{si } n_0 \neq 0 \\ (m_0', n_0') & \text{si } n_0 = 0 \end{cases}$$

donde $m_0' \leq m_0 - 1$ y $n_0' \leq d_0 + \delta - 1$.

En consecuencia, si $m_0 = 0$, después de n_0 pasos de pseudo-derivaciones, obtenemos que $\operatorname{pdeg}(\operatorname{pder}^{(n_0)}(g)) = (0, 0)$ y no se necesitan más. Si $m_0 > 0$, después de $n_0 + 1$ pseudo-derivaciones, la primera coordenada del pseudo-grado es menor que m_0. Sea $(m_1, n_1) = \operatorname{pdeg}(\operatorname{pder}^{(n_0+1)}(g))$ y d_1 el grado total del polinomio que define a $\operatorname{pder}^{(n_0+1)}(g)$. Luego, $m_1 \leq m_0 - 1$, y por la Observación 3.25, $n_1 \leq d_1 \leq (n_0 + 1)(\delta - 1) + d_0$. Como $n_0 \leq d_0$, obtenemos que $d_1 + 1 \leq (d_0 + 1)\delta$.

Estimaremos ahora la longitud de las codificaciones de Thom de los ceros de un E-polinomio.

Lema 3.35 *Sea $f(x) = F(x, e^{h(x)})$ un E-polinomio y $D = \min\{i \ / \ \operatorname{pder}^{(i+1)}(f) = 0\}$. Si $\deg(F) = d$ y $\deg(h) = \delta \geq 2$, se tiene que $D \leq \delta^d(d + 1)$ y el grado total del polinomio que define $\operatorname{pder}^{(i)}(f)$ está acotado por $\delta^d(d + 1)$ para $1 \leq i \leq D$. Si $\deg(h) = 1$, se tiene que $D \leq d(d+1)$ y el grado total del polinomio que define $\operatorname{pder}^{(i)}(f)$ está acotado por d para todo $1 \leq i \leq D$.*

Demostración. Para estimar D, consideraremos los pseudo-grados de la secuencia de pseudo-derivadas de f y los grados de los polinomios que las definen.

Por la Observación 3.34, dado $f(x) = F(x, e^{h(x)})$, obtenemos una secuencia de pseudo-grados $(m_i, n_i)_{0 \leq i \leq k}$ definida como $(m_0, n_0) = \operatorname{pdeg}(f)$ y, para $i \geq 1$, si $m_{i-1} \neq 0$, $N_{i-1} = \sum_{j=0}^{i-1}(n_j + 1)$ y

$$(m_i, n_i) = \operatorname{pdeg}(\operatorname{pder}^{(N_{i-1})}(f)).$$

Si $m_{i-1} = 0$, se tiene que $k = i$, $N_{i-1} = \sum_{j=0}^{i-1}(n_j + 1) - 1$ y $(m_k, n_k) = \operatorname{pdeg}(\operatorname{pder}^{(N_{k-1})}(f)) = (0, 0)$. Sea d_i es el grado total del polinomio que define a $\operatorname{pder}^{(N_{i-1})}(f)$ para $0 \leq i \leq k$.

Por las cotas vistas en la Observación 3.34, $m_i \leq m_{i-1} - 1$, $n_i \leq d_i \leq d_{i-1}\delta + \delta - 1$ y, por lo tanto, $d_i + 1 \leq \delta^i(d+1)$, para todo $i \geq 1$. En particular, $k \leq m_0 = \deg_Y(F) = d$ y $D = N_{k-1}$. Obtenemos entonces que

$$N_{k-1} \leq \sum_{j=0}^{k-1}(n_j+1) \leq \sum_{j=0}^{k-1}(d_j+1) \leq \sum_{j=0}^{k-1}\delta^j(d+1) = \begin{cases} \dfrac{\delta^k-1}{\delta-1}(d+1) & \text{si } \delta \geq 2 \\ k(d+1) & \text{si } \delta = 1, \end{cases}$$

de donde se sigue el resultado. □

Aplicando la Proposición 3.18 y el resultado anterior se deduce lo siguiente:

Proposición 3.36 *Dado un E-polinomio $f(x) = F(x, e^{h(x)})$, donde $F \in \mathbb{Z}[X,Y]$ y $h \in \mathbb{Z}[X]$ son polinomios con grado acotado por d y altura acotada por H, la complejidad de calcular las codificaciones de Thom de todos los ceros de f es de orden $(2dH)^{d^{O(d)}}$. Si $\deg(h) = 1$, la complejidad es de orden $(2dH)^{d^{O(1)}}$.*

3.5.4. Teorema de Budan-Fourier para E-polinomios

En esta subsección daremos una generalización del Teorema de Budan-Fourier válido para polinomios (ver, por ejemplo, [3, Theorem 2.46]) a los E-polinomios. Otra generalización de este teorema puede hallarse en [6].

Comenzaremos introduciendo notación y luego enunciando un lema previo:

Notación 3.37 *Dados un E-polinomio f y $a, b \in \mathbb{R}$, $a < b$, si $\mathrm{PDer}(f) = \{\mathrm{pder}^{(i)}(f)\}_{0 \leq i \leq D}$ es la secuencia de todas las pseudo-derivadas no nulas de f, notaremos*

$$V(\mathrm{PDer}(f), a, b)$$

a la cantidad de cambios de signo de la secuencia $(\mathrm{pder}^{(0)}(f)(a), \ldots, \mathrm{pder}^{(D)}(f)(a))$ menos la cantidad de cambios de signo de la secuencia $(\mathrm{pder}^{(0)}(f)(b), \ldots, \mathrm{pder}^{(D)}(f)(b))$.

Lema 3.38 *Sea $f(x) = F(x, e^{h(x)})$ un E-polinomio no constante definido por $F \in \mathbb{Z}[X,Y]$, con $h \in \mathbb{Z}[X]$. Sean $I = [a,b]$ y $c \in (a,b)$ tales que*

$$\mathrm{pder}^{(i)}(f)(x) = 0 \text{ para algún } x \in I, \text{ para algún } 0 \leq i \leq D \implies x = c.$$

Sea $\mu := \mathrm{mult}(c, f) \geq 0$. Entonces

$$\begin{cases} V\big(\mathrm{PDer}(f), c, b\big) = 0 \text{ y} \\ V\big(\mathrm{PDer}(f), a, c\big) - \mu \geq 0 \text{ y es un número par.} \end{cases}$$

Demostración. Lo probaremos por inducción en el $\mathrm{pdeg}(f)$, considerando el orden lexicográfico. Supongamos que $m := \deg_Y(F)$ y $n := \deg(F(X,0))$ o 0 si $F(X,0) = 0$.

Si $m = 0$, F es un polinomio de una variable y en este caso el resultado vale por [3, Theorem 2.46].

Supongamos que el resultado vale para todo E-polinomio con pseudo-grado menor que (m,n), con $m>0$. Veamos que vale para f, un E-polinomio de pseudo-grado (m,n).

Por el Lema 3.27, como $f\notin\mathbb{R}$, se tiene que $\operatorname{pdeg}(\operatorname{pder}(f))<_{\text{lex}}\operatorname{pdeg}(f)$. Luego, por hipótesis inductiva aplicada a $\operatorname{pder}(f)$, si notamos $\nu=\operatorname{mult}(c,\operatorname{pder}(f))\geq 0$, se tiene que

$$\begin{cases} V\big(\operatorname{PDer}(\operatorname{pder}(f)),c,b\big)=0 \text{ y}\\ V\big(\operatorname{PDer}(\operatorname{pder}(f)),a,c\big)-\nu=2j \text{ para algún } j\in\mathbb{Z},j\geq 0.\end{cases} \tag{3.9}$$

Además, para $i=1\ldots,D$, se verifica que:

i) Por la Observación 3.26, $\operatorname{sg}\Big(\big(\operatorname{pder}^{(i-1)}(f)\big)'(x)\Big)=\operatorname{sg}\big(\operatorname{pder}^{(i)}(f)(x)\big)\ \ \forall\, x\in\mathbb{R}$.

ii) El signo de $\operatorname{pder}^{(i)}(f)$ es constante y no nulo en cada uno de los intervalos $[a,c)$ y $(c,b]$. En particular, si $\operatorname{pder}^{(i)}(f)(c)\neq 0$ entonces $\operatorname{sg}\big(\operatorname{pder}^{(i)}(f)\big)$ es constante y no nulo en $[a,b]$.

iii) Si $\operatorname{pder}^{(i)}(f)(c)=0$, $i<D$, se tiene que:

$$\begin{cases} \operatorname{sg}\big(\operatorname{pder}^{(i)}(f)\big)=\operatorname{sg}\big(\operatorname{pder}^{(i+1)}(f)\big) & \text{en } (c,b]\\ \operatorname{sg}\big(\operatorname{pder}^{(i)}(f)\big)=-\operatorname{sg}\big(\operatorname{pder}^{(i+1)}(f)\big) & \text{en } [a,c).\end{cases}$$

Esto se sigue de la condición $\operatorname{sg}\big(\operatorname{pder}^{(i+1)}(f)\big)$ constante y no nulo en $[a,c)$ y en $(c,b]$ (por ii)), lo cual implica que $\operatorname{pder}^{(i)}(f)$ es estrictamente monótona en $[a,c)$ y en $(c,b]$ (por i)). Como se anula en c, se sigue la afirmación.

Consideremos los siguientes casos:

- Si $f(c)=0$, $\mu\geq 1$ y, por el Lema 3.28, $\nu=\mu-1$. Por iii) aplicado a $i=0$ se tiene que

$$\operatorname{sg}\big(\operatorname{pder}(f)(b)\big)=\operatorname{sg}\big(f(b)\big)=\sigma \text{ y } \operatorname{sg}\big(\operatorname{pder}(f)(a)\big)=-\operatorname{sg}\big(f(a)\big)=-\beta.$$

Obtenemos de esta forma la siguiente situación, para $\sigma,\beta\in\{1,-1\}$:

	a	c	b
f	$-\beta$	0	σ
$\operatorname{pder}(f)$	β		σ

Luego,

$$\begin{cases} V\big(\operatorname{PDer}(f),c,b\big)=0 \text{ y}\\ V\big(\operatorname{PDer}(f),a,c\big)-\mu=1+\nu+2j-\mu=2j\geq 0,\end{cases}$$

como se quería probar.

- Si $f(c)\neq 0$, $\mu=0$. Consideremos los siguientes casos:

  - Si $\nu\geq 1$, $\operatorname{pder}^{(i)}(f)(c)=0$ para $i=1,\ldots,\nu$. Luego, por iii), se tiene que $\operatorname{sg}\big(\operatorname{pder}^{(i)}(f)(b)\big)=\operatorname{sg}\big(\operatorname{pder}^{(\nu+1)}(f)(b)\big)=\sigma$, para todo $i=1,\ldots,\nu$. Además, por ii), $\sigma=\operatorname{sg}\big(\operatorname{pder}^{(\nu+1)}(f)\big)\neq 0$ en $[a,b]$. Por otro lado, iii) implica que los signos de $\operatorname{pder}^{(i)}(f)(a)$ alternan para $i=1,\ldots,\nu+1$.

- Si ν impar, se tiene que $\mathrm{sg}\big(\mathrm{pder}^{(\nu+1)}(f)(a)\big) = -\mathrm{sg}\big(\mathrm{pder}(f)(a)\big)$. Supongamos que $\beta = \mathrm{sg}(f) \neq 0$ en $[a, b]$. Obtenemos que:

	a	c	b
f	β	β	β
$\mathrm{pder}(f)$	$-\sigma$	0	σ
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathrm{pder}^{(\nu)}(f)$	$-\sigma$	0	σ
$\mathrm{pder}^{(\nu+1)}(f)$	σ	σ	σ

  Si $\sigma = \beta$,

$$V\big(\mathrm{PDer}(f), a, c\big) - \mu = V\big(\mathrm{PDer}(\mathrm{pder}(f)), a, c\big) + 1 = 2j + \nu + 1 \geq 0$$

  y resulta ser un número par. Por otro lado,

$$V\big(\mathrm{PDer}(f), c, b\big) = 0 + V\big(\mathrm{PDer}(\mathrm{pder}(f)), c, b\big) = 0.$$

  Si $\sigma = -\beta$,

$$V\big(\mathrm{PDer}(f), a, c\big) - \mu = -1 + V\big(\mathrm{PDer}(\mathrm{pder}(f)), a, c\big) = -1 + 2j + \nu \geq 0$$

  y resulta ser un número par. Por otro lado,

$$V\big(\mathrm{PDer}(f), c, b\big) = 1 - 1 + V\big(\mathrm{PDer}(\mathrm{pder}(f)), c, b\big) = 0.$$

- Si ν es par, se tiene que $\mathrm{sg}\big(\mathrm{pder}^{(\nu+1)}(f)(a)\big) = \mathrm{sg}\big(\mathrm{pder}(f)(a)\big)$. Supongamos que $\beta = \mathrm{sg}(f) \neq 0$ en $[a, b]$. Obtenemos la tabla:

	a	c	b
f	β	β	β
$\mathrm{pder}(f)$	σ	0	σ
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathrm{pder}^{(\nu)}(f)$	$-\sigma$	0	σ
$\mathrm{pder}^{(\nu+1)}(f)$	σ	σ	σ

  Luego, $V\big(\mathrm{PDer}(f), a, c\big) - \mu = V\big(\mathrm{PDer}(\mathrm{pder}(f)), a, c\big) = 2j + \nu \geq 0$ y resulta ser un número par.
  Por otro lado, $V\big(\mathrm{PDer}(f), c, b\big) = V\big(\mathrm{PDer}(\mathrm{pder}(f)), c, b\big) = 0$.

- Si $\nu = 0$, se tiene que $\sigma = \mathrm{sg}\big(\mathrm{pder}(f)(c)\big) \neq 0$ y $\beta = \mathrm{sg}\big(f(c)\big) \neq 0$. De esta forma, obtenemos la tabla:

	a	c	b
f	β	β	β
$\mathrm{pder}(f)$	σ	σ	σ

Sea σ igual o no a β, se obtiene que

$$V\big(\mathrm{PDer}(f), a, c\big) - \mu = V\big(\mathrm{PDer}(\mathrm{pder}(f)), a, c\big) = 2j + \nu \geq 0$$

y resulta ser un número par, y que

$$V\big(\mathrm{PDer}(f), c, b\big) = V\big(\mathrm{PDer}(\mathrm{pder}(f)), c, b\big) = 0.$$

□

Ahora estamos en condiciones de enunciar la generalización a E-polinomios del Teorema de Budan-Fourier:

Teorema 3.39 *Sean f un E-polinomio, $I = (a, b]$ un intervalo y $m \geq 0$ la cantidad de ceros de f en I contados con multiplicidad. Entonces*

$$V(\mathrm{PDer}(f), a, b) - m \geq 0 \text{ y es par.}$$

Demostración. Supongamos que $c_1 < \ldots < c_r$ son los ceros en (a, b) de las funciones en $\mathrm{PDer}(f)$. Sean $c_0 = a$, $c_{r+1} = b$ y $\mu_i = \mathrm{mult}(c_i, f)$ para $i = 1, \ldots, r+1$. Tomemos $d_i \in (c_i, c_{i+1})$ para cada $i = 0, \ldots, r$. Dado que en cada intervalo $[d_i, d_{i+1}]$, para $i = 0, \ldots, r-1$, se verifican las hipótesis del Lema 3.38, se tiene que $V(\mathrm{PDer}(f), c_i, d_i) = 0$ y $V(\mathrm{PDer}(f), d_i, c_{i+1}) - \mu_{i+1} = 2j_i \geq 0$ para algún $j_i \in \mathbb{Z}$ no negativo, para todo $i = 0, \ldots, r-1$. Como $\{x \in (c_i, d_i] \ / \ f(x) = 0\} = \emptyset$ y μ_{i+1} es la cantidad de ceros de f en $(d_i, c_{i+1}]$ contados con multiplicidad, para todo $i = 0, \ldots, r$, se tiene que

$$m = \sum_{i=0}^{r} \mu_{i+1} = \sum_{i=0}^{r} V(\mathrm{PDer}(f), c_i, d_i) + V(\mathrm{PDer}(f), d_i, c_{i+1}) - 2j_i = V(\mathrm{PDer}(f), a, b) - 2j,$$

para $j \in \mathbb{Z}$ no negativo, lo que prueba el resultado. □

Bibliografía

[1] Achatz, Melanie; McCallum, Scott; Weispfenning, Volker. *Deciding polynomial-exponential problems.* Proceedings ISSAC 2008, 215–222, ACM, New York, 2008.

[2] Anai, Hirokazu; Weispfenning, Volker. *Deciding linear-trigonometric problems.* Proceedings ISSAC 2000 (St. Andrews), 14–22, ACM, New York, 2000.

[3] Basu, Saugata; Pollack, Richard; Roy, Marie-Françoise. *Algorithms in real algebraic geometry.* Second edition. Algorithms and Computation in Mathematics, 10. Springer-Verlag, Berlin, 2006. Online version available at https://perso.univ-rennes1.fr/marie-francoise.roy/bpr-ed2-posted3.pdf

[4] Bochnak, Jacek; Coste, Michel; Roy, Marie-Françoise. *Real Algebraic Geometry.* Springer-Verlag, Berlin Heindelberg, 1998.

[5] Canny, John. *Improved algorithms for sign determination and existencial quantifier elimination.* Computer Science Division, University of California, Berkeley 94720 USA. The Computer Journal - Vol. 36, No 5, 1993.

[6] Coste, Michel; Lajous, Tomás; Lombardi, Henri; Roy, Marie-Françoise. *Generalized Budan-Fourier theorem and virtual roots*, J. Complexity 21 (2005), 479–486.

[7] Gabrielov, Andrei; Vorobjov, Nicolai. *Complexity of computations with Pfaffian and Noetherian functions.* In Normal Forms, Bifurcations and Finiteness Problems in Differential Equations, Kluwer, 2004.

[8] Gabrielov, Andrei; Vorobjov, Nicolai; Zell, Thierry. *Betti numbers of semialgebraic and sub-Pfaffian sets.* J. London Math. Soc. (2) 69 (2004), no. 1, 27–43.

[9] Heindel, Lee E. *Integer arithmetic algorithms for polynomial real zero determination.* J. Assoc. Comput. Mach. 18 (1971), 533–548.

[10] Jacobi, Carl G. J. *De eliminatione variabilis e duabus aequationibus algebraicis.* J. Reine Angew. Math. 15 (1836), 101-124.

[11] Khovanskii, Askold. *Fewnomials.* Translations of Mathematical Monographs, 88. American Mathematical Society, Providence, RI, 1991.

[12] Khovanskii, Askold. *On a class of systems of transcendental equations.* Soviet Math. Dokl. 22 (1980), 762–765.

[13] Macintyre, Angus; Wilkie, Alex J. *On the decidability of the real exponential field.* Kreiseliana: About and around Georg Kreisel, A.K. Peters, 1996, pp. 441–467.

[14] Mahler, Kurt. *On some inequalities for polynomials in several variables.* J. London Math. Soc.(2)37 (1962), 341–344.

[15] Maignan, Aude. *Solving one and two-dimensional exponential polynomial systems.* Proc. ISSAC'98, New York, NY: ACM Press (1998), 215-221.

[16] McCallum, Scott; Weispfenning, Volker. *Deciding polynomial-transcendental problems.* J. Symbolic Comput. 47 (2013), 16–31.

[17] Mignotte, Maurice; Stefanescu, Doru. *Polynomials. An Algorithmic Approach.* Springer-Verlag, Singapore, 1999.

[18] Perrucci, Daniel. *Linear solving for sign determination.* Theoret. Comput. Sci 412 (2011) No. 35, 4715-4720.

[19] Richardson, Daniel. *Towards computing non algebraic cylindrical decompositions.* In: S.M.Watt (ed.), Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation, Bonn, Germany, 1991, pp. 247–255.

[20] Roy, Marie-Françoise; Vorobjov, Nicolai. *Finding irreducible components of some real transcendental varieties.* Comput. Complexity 4 (1994), 107–132.

[21] Sylvester, James J. *A method of determining by mere inspection the derivatives from two equations of any degree.* Philos. Mag. 16 (1840), 132-135. Also appears in the Collected Mathematical Papers of James Joseph Sylvester, Vol. 1, Chelsea Publishing Co. (1973), 54-57.

[22] Sylvester, James J. *On rational derivation from equations of coexistence, that is to say, a new and extended theory of elimination.* Philos. Mag. 15 (1839), 428-435. Also appears in the Collected Mathematical Papers of James Joseph Sylvester, Vol. 1, Chelsea Publishing Co. (1973), 40–46.

[23] Tarski, Alfred. *A decision method for elementary algebra and geometry.* 2nd edition. Universtiy of California Press, Berkeley, Los Angeles.

[24] van den Dries, Lou. *Remarks on Tarski's problem concerning* $(\mathbf{R}, +, \cdot, exp)$. Logic Colloquium '82 (Florence, 1982), 97–121, Stud. Logic Found. Math., 112, North-Holland, Amsterdam, 1984.

[25] van den Dries, Lou. *Exponential rings, exponential polynomials and exponential functions.* Pacific Journal of Mathematics. Vol. 113 (1984), no. 1, 51–66.

[26] von zur Gathen, Joachim; Gerhard, Jürgen. *Modern computer algebra.* Second edition. Cambridge University Press, Cambridge, 2003.

[27] Vorobjov, Nikolaj N. (Jr.) *The complexity of deciding consistency of systems of polynomials in exponent inequalities.* J. Symbolic Comput. 13 (1992), no. 2, 139–173.

[28] Waldschmidt, Michael. *Simultaneous aproximation of numbers connected with the exponential function.* J. Austral Maths. Soc. (Serie A) 25 (1978), 466-478.

[29] Weispfenning, Volker. *Deciding linear-transcendental problems.* Computer algebra in scientific computing (Samarkand, 2000), 423–437, Springer, Berlin, 2000.

[30] Wolter, Helmut. *On the "problem of the last root" for exponential terms.* Z. Math. Logik Grundlag. Math. 31 (1985), no. 2, 163–168.

[31] Wolter, Helmut. *On roots of exponential terms.* Math. Logic Quart. 39 (1993), no. 1, 96–102.

[32] Xu, Ming; Li, Zhi-Bin; Yang, Lu. *Quantifier elimination for a class of exponential polynomial formulas.* J. Symbolic Comput. 68 (2015), part 1, 146–168.

[33] Zell, Thierry. *Quantitative study of semi-Pfaffian sets.* PhD thesis, 2003. Disponible en https://arxiv.org/pdf/math/0401079