Tesis de Posgrado

Un algoritmo efectivo para la eliminación de cuantificadores

Puddu, Susana Isabel

1995

Tesis presentada para obtener el grado de Doctor en Ciencias Matemáticas de la Universidad de Buenos Aires

UNIVERSIDAD DE BUENOS AIRES

FACULTAD DE CIENCIAS EXACTAS Y NATURALES

TEMA DE TESIS

Un algoritmo efectivo para la eliminación de cuantificadores

AUTORA

Susana Isabel Puddu

DIRECTOR

Dr. Joos U. Heintz

LUGAR DE TRABAJO

Departamento de Matemática
Facultad de Ciencias Exactas y Naturales (UBA)

Tesis presentada para optar al título de

Doctora en Ciencias Matemáticas

- 1995 -

AGRADECIMIENTOS

A mi director Joos Heintz, por haberme sugerido el tema de esta tesis, por su generosidad y por sus invaluables consejos.

A Juan Sabia, por muchas cosas, pero sobre todo por su ilimitada paciencia en las discusiones compartidas.

A Teresa Krick y Pablo Solernó, por su colaboración.

A mis hijos, por permitirme dedicar más que numerables fines de semana al trabajo y aún así demostrarme su cariño protestando ampliamente por ello.

INDICE

RESUMEN

En este trabajo se construye un algoritmo efectivo para la eliminación de cuantificadores sobre un cuerpo algebraicamente cerrado: Se demuestra que si k es un dominio íntegro, infinito, efectivo y cerrado para la extracción de raíces p-ésimas cuando $car(k)=p>0$ y φ es una fórmula prenexa con r bloques de cuantificadores que involucra a s polinomios $F_1,\ldots,F_s \in k[X_1,\ldots,X_n]$ entonces existe un algoritmo bien paralelizable y sin divisiones de complejidad secuencial del orden de $O(|\varphi|) + D^{(O(n))^r}$ que encuentra una fórmula equivalente a φ libre de cuantificadores, donde $|\varphi|$ es la longitud de φ y $D = \text{máx}\ \{1 + \sum_{i=1}^{s} \deg F_i\ ,\ n\ ,\ s\}$.

Este algoritmo mejora las cotas conocidas que son del orden de $O(|\varphi|) + D^{n^{cr}}$ con $c > 1$ una constante universal (ver [Ie, 1989] y [Fi-Ga-Mo, 1990]). En particular se obtiene que el carácter doblemente exponencial de las cotas conocidas en el caso de un solo bloque de cuantificadores (del tipo $O(|\varphi|) + D^{n^c}$ con $c > 1$) no es intrínseco, es decir no depende del problema sino de los algoritmos y del tipo de codificación utilizados.

Los resultados obtenidos se basan fundamentalmente en el cambio del modelo de codificación de los polinomios: en vez de representar los polinomios de salida por el vector de sus coeficientes, se los codifica por medio de una red aritmética sin comparaciones ni ramas que permite evaluarlos en cualquier punto.

Como aplicación, se calcula la Forma de Chow de una variedad proyectiva irreducible.

1. Introducción

Sea k un cuerpo arbitrario y sea $\overline{k}$ un cuerpo algebraicamente cerrado que lo contiene. Se denota por $\mathcal{L}(k)$ al lenguaje de primer orden sobre el cuerpo $\overline{k}$ con constantes en k. En otras palabras, $\mathcal{L}(k)$ es el conjunto de fórmulas que son combinaciones booleanas $(\wedge,\vee,\neg)$ de polinomios en varias variables a coeficientes en k, igualados a cero, admitiéndose también los cuantificadores $\exists$ y $\forall$ para modificar únicamente a elementos de $\overline{k}$, representados por variables.

Es un hecho clásico de la teoría de modelos que el lenguaje de primer orden de los cuerpos algebraicamente cerrados admite eliminación de cuantificadores, es decir, para toda fórmula $\phi \in \mathcal{L}(k)$ existe una fórmula $\psi \in \mathcal{L}(k)$, sin cuantificadores, que describe el mismo subconjunto de $\overline{k}^r$, donde r es el número de variables de ϕ que no están afectadas por ningún cuantificador.

Muchos problemas geométricos y algebraicos pueden ser formulados en el lenguaje de primer orden de los cuerpos algebraicamente cerrados y su solución consiste, precisamente, en eliminar los cuantificadores. Por esta causa, es de particular interés encontrar algoritmos que realicen esta eliminación (i.e., que a partir de la fórmula ϕ encuentren la fórmula ψ).

En [He-Wü, 1975] J. Heintz y R. Wüthrich presentaron por primera vez un algoritmo de eliminación de cuantificadores con cotas de complejidad para cuerpos algebraicamente cerrados de característica dada. En realidad, parece ser que ya hacia 1940, A. Tarski

conocía la existencia de algoritmos para este problema, aunque no los describió explícitamente (ver [Tar, 1951]). De ahí en más, se ha intentado exhibir algoritmos de eliminación de cuantificadores cada vez más eficientes.

Los algoritmos serán representados como redes aritméticas, es decir por medio de grafos orientados, donde en cada nodo se realiza una operación aritmética (suma, resta, producto) o booleana, una comparación de elementos de $\overline{k}$ o una selección. Por lo tanto, la noción de eficiencia de un algoritmo se puede medir en términos del grafo correspondiente.

Así aparecen dos nociones básicas de *complejidad* de algoritmos:

· la *complejidad secuencial*: el número de nodos del grafo

· la *complejidad paralela*: la longitud del camino más largo en el grafo.

Dada una fórmula $\phi \in \mathcal{L}(k)$, sean

$|\phi|$ = longitud de ϕ = número de símbolos necesarios para codificar a ϕ

n = número de variables que aparecen en ϕ

$D = 2 + \sum_{i=1}^{s} gr\ F_i$, donde $F_1, \ldots, F_s \in k[X_1,\ldots,X_n]$ son todos los polinomios que aparecen en ϕ

y, en el caso en que ϕ sea prenexa (i.e., cuando todos los cuantificadores aparecen al principio de ϕ), sea

r = número de bloques de cuantificadores distintos en ϕ

En [He-Wü, 1975], [Wü, 1977] y [He, 1983], Heintz y Wüthrich desarrollaron algoritmos para la eliminación de cuantificadores en

un cuerpo algebraicamente cerrado, cuyas cotas superiores de complejidad secuencial son doblemente exponenciales del tipo $D^{n^{cn}}|\phi|$, donde c es una constante universal.

Más tarde, basándose en técnicas fundamentales de [Ch-Gr, 1983] y [He, 1983], Chistov y Grigor'ev consideran el problema para fórmulas prenexas y obtienen en [Ch-Gr, 1984] y [Gr, 1987], cotas secuenciales más precisas, a saber, $D^{n^{cr}}|\phi|$, donde c es una constante universal mayor o igual que 2. Como se ve, el carácter doblemente exponencial de la cota depende únicamente del número r de bloques de cuantificadores de ϕ. Sin embargo, la complejidad del algoritmo depende de propiedades aritméticas del cuerpo de constantes k, hecho que se debe a la utilización de subalgoritmos de factorización de polinomios.

Cabe señalar también que ninguno de los algoritmos mencionados arroja buenas cotas en paralelo (es decir, cotas del orden del cuadrado del logaritmo en base 2 de las cotas secuenciales).

Finalmente, combinando los métodos de [He, 1983] con versiones efectivas del Nullstellensatz (ver [Ko, 1988], [Ca-Gu-Gu, 1991], [Ca, 1989], [Ph, 1988], [Am, 1989] y [Br, 1989], que afinaron los resultados de [Br, 1987], [Ca-Ga-He, 1988], [Ca-Ga-He, 1989] y [Be-Yg, 1991]), en [Fi-Ga-Mo, 1990] se construye un algoritmo de eliminación de cuantificadores con las mismas cotas secuenciales obtenidas en [Ch-Gr, 1984] y [Gr, 1987], pero con la ventaja de que se obtienen cotas del orden de $n^{cr} D^c + c \log_2(|\phi|)$ en paralelo y además se evita que la complejidad dependa de propiedades parti-

culares del cuerpo de constantes k. Más tarde, el mismo resultado fue obtenido por Ierardi ([Ie, 1989]). Una consecuencia importante y directa de la paralelización es que la eliminación de cuantificadores es posible en EXPSPACE (ver [Bo, 1977], [Bo et al, 1982] y [Ma-Tu, 1995]).

En todos estos algoritmos, los polinomios están codificados en forma densa (i.e., están representados por el vector de todos sus coeficientes, aún los nulos, siguiendo un orden preestablecido de los monomios) y, en el modelo de la representación densa de los polinomios, las cotas superiores obtenidas en [Fi-Ga-Mo, 1990] son optimales como medida general de complejidad, no sólo desde el punto de vista de la complejidad secuencial sino también de la paralela.

Por consiguiente, los caracteres doblemente exponencial y simplemente exponencial de dichas cotas son intrínsecos del problema, siempre que se adopte la representación densa de los polinomios como modelo. Esto muestra que, para intentar mejorar las cotas, es inevitable modificar la estructura de datos para los polinomios.

Una forma de codificación de polinomios que mostró ser efectiva para conseguir algoritmos más eficientes en problemas de álgebra y geometría es la utilización de *straight line programs*: circuitos aritméticos que no contienen ramas ni comparaciones y que permiten evaluar los polinomios en cualquier punto (ver, por ejemplo, [He-Si, 1981], [Kal, 1988], [Gi-He, 1993], [Gi-He-Sa, 1993], [Kr-Par, 1994], [Fi-Gi-Smi, 1995] o [Gi et al, 1995]).

En este trabajo se obtiene un algoritmo rápido de eliminación de cuantificadores, utilizando los resultados de [Gi-He, 1993] para el cálculo de la dimensión de una variedad algebraica afín. Para ello, los polinomios serán codificados, según convenga, a veces en forma densa, a veces por medio de un straight line program y a veces combinando ambas formas (es decir, en forma densa con respecto a algunas variables y codificando sus coeficientes en las restantes por medio de un straight line program).

La construcción del algoritmo obtenido en este trabajo se hace en varias etapas.

En un primer paso se consideran fórmulas del tipo:

$$\exists\, x_{n-m+1},\ldots,\exists\, x_n : F_1(x_1,\ldots,x_n) = 0 \wedge \ldots \wedge F_s(x_1,\ldots,x_n) = 0$$

donde $F_1,\ldots,F_s$ son polinomios en n variables dados por un straight line program de longitud L. Sea $d \geq m$ una cota para los grados de los polinomios $F_1,\ldots,F_s$ respecto de las m variables a eliminar.

El algoritmo de eliminación obtenido en este caso tiene una complejidad secuencial del orden de $L.d^{O(m)} + s^{O(1)}.d^{O(m)}$, lo que mejora sustancialmente las cotas existentes. Para probarlo se exhibe un ejemplo que muestra que, aún con un solo bloque de cuantificadores, cualquier algoritmo que utilice el modelo de representación densa de polinomios tendrá una complejidad no menor a d^{m^2}.

El caso de un solo bloque de cuantificadores existenciales es de particular importancia ya que aparece con mucha frecuencia en la formulación de problemas geométricos y algebraicos (por ejemplo,

en la definición de la Forma de Chow de una variedad proyectiva equidimensional o en el problema de la pertenencia de un polinomio a un ideal).

Luego se modifica de manera conveniente el algoritmo obtenido anteriormente para permitir la inclusión de desigualdades, es decir, se consideran fórmulas del tipo:

$$\exists\, x_{n-m+1},\ldots,\exists\, x_n : F_1(x_1,\ldots,x_n) = 0 \wedge \ldots \wedge F_s(x_1,\ldots,x_n) = 0 \wedge$$
$$\wedge\, G_1(x_1,\ldots,x_n) \neq 0 \wedge \ldots \wedge G_{s'}(x_1,\ldots,x_n) \neq 0$$

donde $F_1,\ldots,F_s,G_1,\ldots,G_{s'}$ son polinomios en n variables y se obtiene un algoritmo de complejidad secuencial $L^2(s.s'.\delta)^{O(1)}\ d^{O(m)}$, donde δ es una cota para los grados de los polinomios $G_1,\ldots,G_{s'}$ en las variables a eliminar, $d \geq m$ es una cota para los grados de los polinomios $F_1,\ldots,F_s$ también con respecto a las variables a eliminar y L es la longitud del straight line program que codifica a $F_1,\ldots,F_s,G_1,\ldots,G_{s'}$.

También en este caso las cotas obtenidas mejoran la complejidad de cualquier algoritmo de eliminación que sólo utilice la representación densa de polinomios.

Dado que el cuantificador existencial conmuta con las disjunciones, el algoritmo obtenido resuelve el problema de la eliminación para cualquier fórmula prenexa con un solo bloque de cuantificadores existenciales escrita en forma disyuntiva (es decir, un solo bloque de cuantificadores existenciales precediendo a una disjunción de conjunciones de igualdades y desigualdades de polinomios). Para terminar de resolver el problema de la eliminación

en el caso en que haya un solo bloque de cuantificadores, teniendo en cuenta que mediante la negación se pueden transformar los cuantificadores universales en existenciales ($\forall = \neg\, \exists\, \neg$), se da un algoritmo que permite encontrar, a partir de una fórmula que no contenga cuantificadores, otra equivalente que sea una disjunción de conjunciones de igualdades y desigualdades polinomiales para poder reducirse luego al caso anterior.

De este modo se obtiene un algoritmo que resuelve el problema de la eliminación en el caso de una fórmula prenexa con un solo bloque de cuantifi- cadores. La complejidad secuencial de este algoritmo mejora la de cualquier algoritmo en el modelo de la representación densa de polinomios.

Luego, basándose en el caso de un solo bloque de cuantificadores, se obtiene el siguiente resultado:

Sea φ una fórmula prenexa con r bloques de cuantificadores que involucra a s polinomios $F_1,\ldots,F_s \in k[X_1,\ldots,X_n]$ *codificados en forma densa, sea* $D = \max\{1+\sum_{i=1}^{s} \deg F_i,\ n,\ s\}$ *y sea* $|\varphi|$ *la longitud de φ. A menos de una preparación previa existe un algoritmo bien paralelizable y sin divisiones de complejidad secuencial del orden de* $O(|\varphi|) + D^{(O(n))^r}$ *que encuentra una fórmula equivalente a φ, libre de cuantificadores.*

Por último, como aplicación de los algoritmos de eliminación exhibidos anteriormente, se construye un algoritmo que calcula la Forma de Chow de una variedad proyectiva irreducible. Las cotas de

complejidad logradas de esta manera mejoran los resultados conocidos (ver, por ejemplo, [Ca, 1990]).

Todos los algoritmos exhibidos en este trabajo son bien paralelizables (es decir, con complejidad paralela del orden del cuadrado del logaritmo en base 2 de la complejidad secuencial) y no contienen divisiones. Además, son no uniformes ya que para su construcción es necesario un preprocesamiento cuyo costo excede las clases de complejidad consideradas en este trabajo. Sin embargo, dicho prepocesamiento, que consiste en la elección de ciertos números, puede ser reemplazado por una elección aleatoria con una probabilidad de error baja. En este sentido, todos los algoritmos aquí construidos son uniformes con el mismo orden de complejidad promedio si se los considera aleatorios.

2. Preliminares

2.1. Lenguaje de primer orden de un cuerpo algebraicamente cerrado

Sea k un cuerpo y sea $\overline{k}$ un cuerpo algebraicamente cerrado que lo contiene. El *lenguaje de primer orden* (o *lenguaje elemental*) sobre $\overline{k}$ con constantes en k, denotado por $\mathcal{L}(k)$, se define de la siguiente manera: los símbolos no lógicos de $\mathcal{L}(k)$ son $\{a,\ a\in k\}$, $+,-,\cdot,=$, las variables son indeterminadas $X_1,\ldots,X_n,\ldots$ sobre $\overline{k}$ y los términos son polinomios en varias variables a coeficientes en k. Un término típico tiene la forma $F \in k[X_1,\ldots,X_n]$ y una fórmula atómica es $F = 0$ (o $F \neq 0$ para su negación). El lenguaje $\mathcal{L}(k)$ se construye a partir de estas fórmulas utilizando los conectivos lógicos $\wedge$, $\vee$, $\neg$ y los cuantificadores $\exists$ y $\forall$ que sólo se pueden aplicar a elementos de $\overline{k}$ (representados por variables) y no a subconjuntos ni a relaciones de $\overline{k}$. $\mathcal{L}(k)$ es un conjunto de palabras finitas (fórmulas) sobre el alfabeto de símbolos de $\mathcal{L}(k)$ (variables, símbolos no lógicos, conectivos lógicos, cuantificadores, paréntesis). Para una definición precisa, ver [Ch-Kei].

Para cada fórmula $\phi \in \mathcal{L}(k)$, se define la *longitud* de ϕ como la cantidad de símbolos necesarios para escribir a ϕ. La longitud de ϕ será denotada por $|\phi|$. Las variables que en la fórmula ϕ aparecen acompañadas por un cuantificador (existencial o universal) se llaman *variables ligadas* o *variables cuantificadas* y las restantes se llaman *variables libres*.

Sean $\phi,\psi \in \mathcal{L}(k)$. Diremos que ϕ y ψ son *equivalentes* con respec-

to a $\bar{k}$ si se verifican las dos condiciones siguientes:

(i) ϕ y ψ tienen el mismo número de variables libres

(ii) Si m es la cantidad de variables libres de ϕ entonces, para todo $x = (x_1,\ldots,x_m) \in \bar{k}^m$, $\phi(x)$ es verdadera si y sólo si $\psi(x)$ lo es (donde $\phi(x)$ significa reemplazar las m variables libres de ϕ por $x_1,\ldots,x_m$).

Diremos que una fórmula $\phi \in \mathcal{L}(k)$ es *prenexa* cuando todos los cuantificadores que aparecen en ella se hallan al principio. Cabe señalar que para toda fórmula $\phi \in \mathcal{L}(k)$ puede encontrarse eventualmente en tiempo lineal, una fórmula prenexa $\psi \in \mathcal{L}(k)$ equivalente a ella, renombrando las variables. Este procedimiento no modifica ni $|\phi|$ ni el grado de los polinomios que aparecen en ϕ, y la cantidad de variables de ψ está acotada por la suma de la cantidad de variables libres y la cantidad de cuantificadores de ϕ.

Es un hecho clásico de la teoría de modelos que el lenguaje de primer orden de los cuerpos algebraicamente cerrados admite *eliminación de cuantificadores*, es decir, para toda fórmula $\phi \in \mathcal{L}(k)$ existe una fórmula $\psi \in \mathcal{L}(k)$, sin cuantificadores, equivalente a ϕ con respecto a todos los cuerpos algebraicamente cerrados que contienen a k. Además, esta eliminación puede hacerse en forma algorítmica. Por lo tanto, si la fórmula original no contiene variables libres, un algoritmo de eliminación efectivo tendrá como salida una combinación booleana de igualdades y desigualdades de elementos de k que es verdadera o falsa. En este caso se dice que se trata de un *problema de decisión*.

Muchos problemas geométricos y algebraicos pueden ser enunciados en términos de fórmulas del lenguaje de primer orden de los cuerpos algebraicamente cerrados y su solución consiste precisamente en eliminar los cuantificadores. Por ejemplo, el problema de determinar si dos polinomios en $k[X]$, de grado n y m respectivamente, tienen un factor común en $k[X]$ es equivalente a que tengan una raíz en común en $\overline{k}$ y por lo tanto puede describirse por medio de la fórmula φ en $\mathcal{L}(\overline{k})$:

$$(\exists\, x) : a_0 + a_1.x + a_2.x^2 + \ldots + a_n.x^n = 0 \wedge$$
$$\wedge\, b_0 + b_1.x + b_2.x^2 + \ldots + b_m.x^m = 0 \wedge a_n \neq 0 \wedge b_m \neq 0$$

Esta es una fórmula prenexa conteniendo 4 polinomios en n+m+3 variables. Las única variable ligada es x, y las variables libres son a_i, b_j $(0 \leq i \leq n,\ 0 \leq j \leq m)$.

Si consideramos la matriz $C \in k^{(n+m)x(n+m)}$

$$C = \begin{pmatrix}
a_n & a_{n-1} & \ldots\ldots & a_0 & 0 & \ldots\ldots\ldots\ldots & 0 \\
0 & a_n & a_{n-1} & \ldots\ldots & a_0 & 0 \ldots\ldots\ldots & 0 \\
\multicolumn{7}{c}{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots} \\
0 & \ldots\ldots\ldots\ldots\ldots & 0 & a_n & a_{n-1} & \ldots\ldots & a_0 \\
b_m & b_{m-1} & \ldots\ldots\ldots\ldots & b_0 & 0 & \ldots\ldots & 0 \\
0 & b_m & b_{m-1} & \ldots\ldots\ldots\ldots & b_0 & 0 \ldots\ldots & 0 \\
\multicolumn{7}{c}{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots} \\
0 & \ldots\ldots & 0 & b_m & b_{m-1} & \ldots\ldots\ldots\ldots & b_0
\end{pmatrix}$$

resulta que $\det C = 0 \wedge a_n \neq 0 \wedge b_m \neq 0$ es una fórmula sin cuantificadores equivalente a φ que soluciona el problema (notar que det C es la resultante entre los polinomios dados).

Los algoritmos efectivos para la eliminación de cuantificadores permiten resolver computacionalmente aquellos problemas que pueden formularse en el lenguaje de primer orden de los cuerpos algebraicamente cerrados.

2.2. Notaciones

Sea k un dominio de integridad infinito. Supondremos que k es *efectivo*, es decir, que las operaciones aritméticas (suma, resta, producto) y las comparaciones entre elementos de k son realizables por un algoritmo. En el caso en que k tenga característica $p>0$, supondremos también que k es cerrado para la extracción de raíces p-ésimas y que la extracción de tales raíces es realizada por un algoritmo.

Sea k' el cuerpo de cocientes de k y sea $\overline{k}$ una clausura algebraica de k'. El espacio afin n-dimensional sobre $\overline{k}$, equipado con su topología de Zariski y su anillo de coordenadas de funciones polinomiales, será denotado por $\mathbb{A}^n(\overline{k})$.

Sean $X_1,\ldots,X_n$ indeterminadas sobre k. Dado $f \in k[X_1,\ldots,X_n]$, deg f denota el grado total de f y $\deg_{X_1,\ldots,X_i} f$ denota el grado parcial de f respecto de $X_1,\ldots,X_i$ $(1\le i\le n)$. Además, si $f_1,\ldots,f_r$ son polinomios en $k[X_1,\ldots,X_n]$, $\mathrm{mcd}_{X_n}(f_1,\ldots f_r)$ denota el máximo común divisor entre $f_1,\ldots f_r$ respecto de la variable X_n (es decir, viendo a $f_1,\ldots f_r$ como polinomios a coeficientes en $k(X_1,\ldots,X_{n-1})$ en la indeterminada X_n)

Si μ y ρ son funciones de $\mathbb{N}$ en $\mathbb{N}$, $\mu(n) = O(\rho(n))$ significará

que existe una constante $c > 0$ independiente de n tal que, para todo $n \in \mathbb{N}$, $\mu(n) \leq c.\rho(n)$.

2.3. El modelo algorítmico

2.3.1. Generalidades

En este trabajo se construirá un algoritmo efectivo de eliminación de cuantificadores. La noción de algoritmo que se va a utilizar es la de una red aritmética, es decir, un grafo orientado acíclico donde cada nodo interno representa o bien una operación aritmética de elementos de k (incluyendo la extracción de raíces p-ésimas en el caso en que car $k = p > 0$), o bien una operación booleana, o bien una comparación entre elementos de k seguida por un selector, y donde los nodos externos representan la entrada y la salida de la red.

También es necesario mencionar que una parte del algoritmo corresponde al manejo puro de las fórmulas. Por ejemplo, de la fórmula de entrada hay que extraer los polinomios que contenga para construir la fórmula de salida. En pocas palabras, se admitirán el mismo tipo de operaciones elementales con los símbolos del lenguaje de primer orden que con los elementos de k (concatenación de palabras, intercambio o inserción de símbolos del lenguaje, etc).

Se definen las dos siguientes nociones de *complejidad* para un algoritmo, en términos de su grafo:

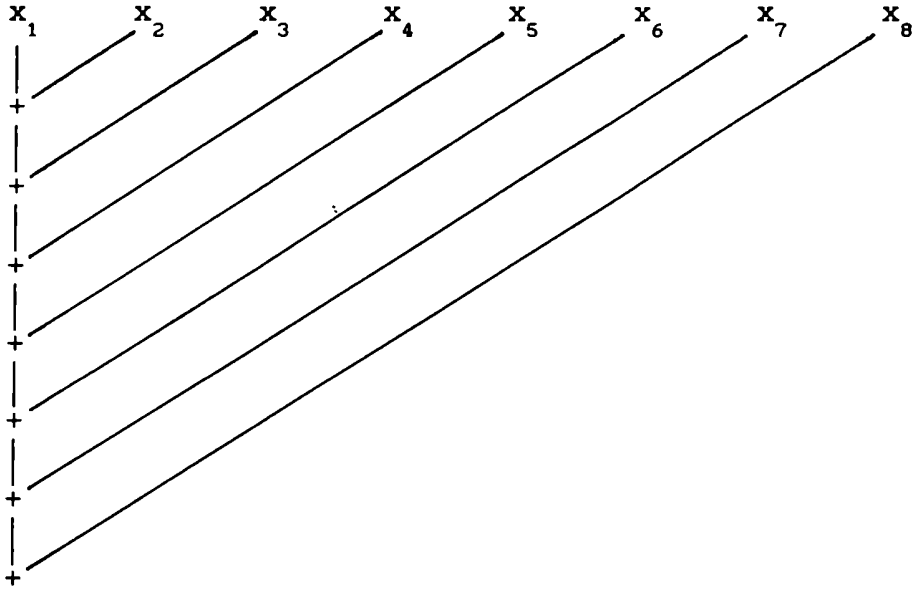
(i) la *complejidad secuencial*: es el número de nodos del grafo,

sin contar los nodos de entrada.

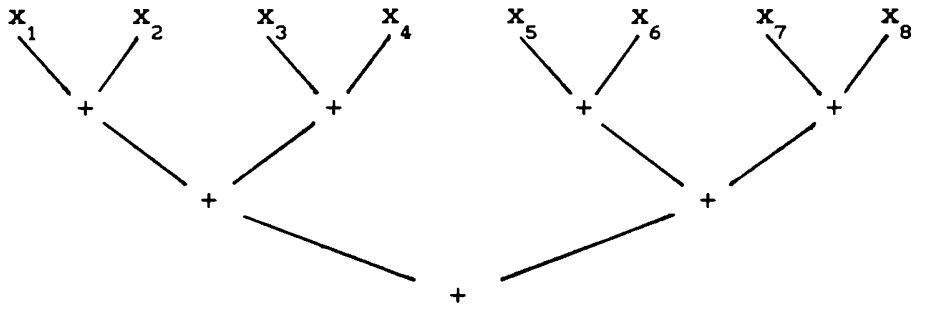
(ii) la *complejidad paralela*: es el número de nodos de una cadena maximal del grafo (es decir, el camino orientado más largo a seguir para llegar a un resultado), sin contar los nodos de entrada.

Si se supone que cada operación se realiza en una unidad de tiempo dada, la complejidad secuencial representa el tiempo necesario para ejecutar el algoritmo utilizando un único procesador, mientras que, dado que la profundidad del grafo refleja el número de operaciones que deben esperar el resultado de operaciones anteriores, la complejidad paralela representa el tiempo mínimo que se necesita para ejecutar el algoritmo si se dispone de una cantidad de procesadores igual a la complejidad secuencial, funcionando en paralelo.

Para ejemplificar,la suma de 8 números $x_1,\ldots,x_8$ puede realizarse por medio del algoritmo:



pero también puede realizarse el mismo cálculo utilizando el algoritmo:



El primer algoritmo tiene complejidad secuencial 7 y complejidad paralela 7, y el segundo tiene complejidad secuencial 7 y complejidad paralela 3.

Diremos que un algoritmo es *bien paralelizable* si su complejidad paralela es del orden del cuadrado del logaritmo en base 2 de la complejidad secuencial.

2.3.2. Codificación de polinomios

Los polinomios que aparezcan en los algoritmos construídos en este trabajo serán codificados, según convenga, en alguna de las siguientes formas:

a) En *forma densa*, es decir, representados por el vector de todos sus coeficientes (aún los nulos) en un orden preestablecido de los monomios, fijado el grado y la cantidad de variables.

b) Por medio de un *straight line program* (también llamado programa de evaluación o circuito aritmético) , es decir, representados por una red aritmética que no hace intervenir comparaciones ni

ramas: un straight line program en $k[X_1,\ldots,X_n]$ sin divisiones es un vector $b = (b_1,\ldots,b_L) \in (k[X_1,\ldots,X_n])^L$ tal que $\forall\ 1 \le \rho \le L$ se cumple alguna de las siguientes condiciones:

- $b_\rho \in \{X_1,\ldots,X_n\}$
- $b_\rho \in k$
- $\exists\ \sigma,\tau < \rho\ /\ b_\rho = b_\tau * b_\sigma$, donde $* \in \{+,-,\cdot\}$

En este caso L se llama la *longitud* del straight line program b. Para mayor precisión sobre estas redes ver [St, 1972], [Gat, 1986], [Sto, 1989] o [He, 1989].

c) Combinando la representación de polinomios en forma densa con straight line programs, es decir, en forma densa con respecto a algunas variables y dando sus coeficientes (que son polinomios en las restantes variables) por medio de un straight line program.

Los polinomios de salida siempre estarán dados por medio de un straight line program lo que, en general, reduce el espacio para codificarlos (y, en consecuencia, la complejidad del algoritmo) tal como se ve en el siguiente ejemplo:

Sea $f \in \mathbb{Z}[X_1,\ldots,X_n]$, $f = (X_1.X_2\ldots X_n)^{2^r}$ con $r \in \mathbb{N}$. Si el grado prefijado es $n.2^r$, preestableciendo un orden para los monomios, la representación en forma densa de f será el vector de $\frac{(n.2^r+ n)!}{(n.2^r)!.n!}$ coordenadas que tiene un 1 en el lugar correspondiente al monomio $X_1^{2^r}.X_2^{2^r}\ldots X_n^{2^r}$ y cero en los restantes lugares. Por lo tanto, esta forma de codificar a f insume más de 2^{rn} lugares de memoria.

Sin embargo, el mismo polinomio f puede ser representado por medio de un straight line program de longitud 2n + r - 1:

$$(X_1, X_2, \ldots, X_n, X_1X_2, X_1X_2X_3, \ldots, X_1X_2\ldots X_n, (X_1X_2\ldots X_n)^2, (X_1X_2\ldots X_n)^{2^2},$$
$$(X_1X_2\ldots X_n)^{2^3}, \ldots, (X_1X_2\ldots X_n)^{2^r})$$

Otra codificación que podría utilizarse en este caso es la forma rala (especificando únicamente los coeficientes no nulos del polinomio y a qué monomio corresponden). La ventaja de los straight line programs con respecto a la forma rala es que los primeros permiten cambios lineales de variables sin modificar sustancialmente la complejidad.

2.4. Herramientas utilizadas

Los algoritmos construidos en este trabajo se basan esencialmente en las técnicas desarrolladas en [Gi-He, 1993] sobre el cálculo de la dimensión de una variedad algebraica a partir de la recuperación de los puntos aislados de una variedad de dimensión cero apropiada.

Las técnicas del álgebra lineal efectiva utilizadas se basan en el algoritmo bien paralelizable de Berkowitz ([Ber, 1984]) que calcula en tiempo polinomial todos los coeficientes del polinomio característico de una matriz cuadrada a coeficientes en un dominio íntegro. Los coeficientes del polinomio característico se representan mediante un straight line program sin divisiones.

Para calcular el rango de una matriz, se combina el algoritmo de Berkowitz con un resultado de Mulmuley ([Mul, 1986]) que realiza el rango como la multiplicidad del cero en el polinomio carac-

terístico de una matriz cuadrada auxiliar.

Para poder aplicar los resultados mencionados será necesario introducir nuevas indeterminadas (parámetros). Estas serán luego eliminadas de la salida utilizando el siguiente teorema de Heintz-Schnorr (ver [He-Sch, 1982]):

Se considera el conjunto $W(D,n,v)$ *de polinomios de* $k[T_1,\ldots,T_n]$ *de grado menor o igual que* D *que pueden ser evaluados por medio de un straight line program de longitud a lo sumo* v. *Sea* Γ *un subconjunto de* k *de cardinal* $2v(1+D)^2$. *Entonces existe un subconjunto* $Q(D,n,v,\Gamma) = \{\gamma_1,\ldots,\gamma_r\}$ *de* Γ^n, *donde* $r = 6(v+n)(v+n+1)$, *que verifica la propiedad siguiente: todo polinomio de* $W(D,n,v)$ *que se anula sobre* $\{\gamma_1,\ldots,\gamma_r\}$ *es idénticamente nulo.*

Un subconjunto $H \subseteq \Gamma^n$ se llamará una *correct test sequence* para D, n y v si cumple:

$$f \in W(D,n,v),\ f(h) = 0 \quad \forall\ h \in H \ \Rightarrow\ f = 0$$

En este sentido, el teorema de Heintz-Schnorr asegura la existencia de correct test sequences incluidas en un subconjunto de cardinal apropiado dado de antemano.

En este punto, el hecho de que los algoritmos construidos en este trabajo no contengan divisiones es fundamental debido a que los polinomios de salida se obtendrán evaluando un straight line program en los puntos de una correct test sequence adecuada.

Aunque la elección de una correct test sequence puede hacerse algorítmicamente, el costo requerido para su construcción excede

las clases de complejidad consideradas en este trabajo. Sin embargo, fijados los parámetros de entrada, esta elección es independiente del problema en sí. Por lo tanto, se pensará que ya se ha obtenido la correct test sequence por medio de un preprocesamiento cuyo costo no será considerado en las cotas de complejidad halladas. En este sentido se dirá que los algoritmos obtenidos son *no uniformes*.

Por otra parte, el Teorema 4.4. en [He-Sch, 1982] garantiza que se puede elegir al azar una correct test sequence con una probabilidad de error que es siempre menor que $\frac{1}{262144}$ y que decrece arbitrariamente a medida que los parámetros aumentan. En consecuencia, los algoritmos construidos son uniformes con el mismo orden de complejidad promedio si se los considera aleatorios.

De este modo, los resultados logrados son válidos no sólo en el sentido del modelo de complejidad no uniforme sino también en el sentido de los algoritmos probabilísticos (se dice que un algoritmo es *probabilístico* cuando su construcción se basa en la elección aleatoria de ciertos números. De esta elección dependerá que la respuesta dada por el algoritmo sea correcta o no. De esta manera, un algoritmo probabilístico trabaja en la forma usual, con la diferencia de que, eventualmente, las decisiones que toma tienen una cierta probabilidad de error asociada. Para mayores detalles ver, por ejemplo, [Bal et al, 1988], [Gi-He, 1993] y [Gi-He-Sa, 1993]).

Cuando la característica p de k sea positiva, será necesario extraer raíces p-ésimas de ciertos elementos en extensiones del

anillo de base. Estas extracciones aparecerán únicamente en subrutinas y no modificarán los resultados finales ni el comportamiento de los algoritmos (ver [Gi-He-Sa, 1993]).

En el caso $k = \mathbb{Z}$, cada nodo de la red aritmética que corresponde a una operación fundamental en el anillo de base $\mathbb{Z}$ puede ser reemplazado por un circuito booleano que procesa bits. Teniendo en cuenta el crecimiento de los coeficientes de los polinomios que aparecen en los resultados intermedios de los algoritmos, las redes aritméticas pueden ser transformadas en redes booleanas del mismo orden de complejidad en forma natural y los resultados siguen siendo válidos para el modelo de complejidad bit de algoritmos representados por redes booleanas, pero este análisis excede los alcances del presente trabajo (para un análisis de este tipo ver, por ejemplo, [Kr-Par, 1994]).

3. Un caso fundamental

En esta sección se exhibirá un algoritmo de eliminación de cuantificadores para fórmulas prenexas que contengan un único bloque de cuantificadores existenciales afectando a una conjunción de igualdades y desigualdades polinomiales. Este algoritmo utiliza convenientemente la codificación de polinomios por medio de straight line programs. En algunos casos, será necesario efectuar cambios de codificación. El método para realizarlos se describe en 3.1. En 3.2. se analizará el caso particular de una fórmula sin desigualdades. Luego, en 3.3. se exhibirá un ejemplo que muestra que las cotas obtenidas mejoran la complejidad de cualquier algoritmo posible en el modelo de la representación densa de polinomios. Finalmente, en 3.4. se adaptará el algoritmo de 3.2. para el caso en que también haya desigualdades.

A lo largo de esta sección se mantendrán las notaciones establecidas en 2.2.

3.1. Cambios de codificación

En la siguiente proposición se construye un algoritmo que codifica en forma densa respecto de algunas variables a un conjunto de polinomios dados por un straight line program.

Proposición 3.1.1. *Sea* m *un número natural tal que* $1 \leq m \leq n$ *y sean* $F_1, \ldots, F_s \in k[X_1, \ldots, X_n]$ *polinomios dados por un straight line program de longitud* ℓ, *cuyos grados totales en las variables*

$X_{n-m+1},\ldots,X_n$ *están acotados por un número natural* $d \geq m$.
Entonces existe un algoritmo bien paralelizable y sin divisiones, de complejidad secuencial $\ell \cdot d^{O(m)}$ *que escribe a los polinomios* $F_1,\ldots,F_s \in k[X_1,\ldots,X_{n-m}][X_{n-m+1},\ldots,X_n]$ *en forma densa en las variables* $X_{n-m+1},\ldots,X_n$, *dando a sus coeficientes (que son elementos de* $k[X_1,\ldots,X_{n-m}]$*) en forma de un straight line program de longitud* $\ell \cdot d^{O(m)}$.

Demostración: se utilizará el método de interpolación para escribir a los polinomios en forma densa en la última variable y luego se iterará el procedimiento m veces.

Sean $\varepsilon_0,\ldots,\varepsilon_d \in k$ d+1 puntos distintos y sea $A \in k^{(d+1)x(d+1)}$ la matriz definida por:

$$A = \begin{pmatrix} 1 & \varepsilon_0 & \varepsilon_0^2 & \ldots. & \varepsilon_0^d \\ 1 & \varepsilon_1 & \varepsilon_1^2 & \ldots. & \varepsilon_1^d \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \varepsilon_d & \varepsilon_d^2 & \ldots. & \varepsilon_d^d \end{pmatrix}$$

El primer paso consiste en calcular A^{-1} (notar que, como los puntos son distintos, la matriz A es inversible). El costo de este paso es $d^{O(1)}$.

Para cada i tal que $1 \leq i \leq s$,

$$F_i = \sum_{j=0}^{d} a_{ij} X_n^j \quad \text{con } a_{ij} \in k[X_1,\ldots,X_{n-1}]$$

Dado un punto $(\xi_1,\ldots,\xi_{n-1}) \in k^{n-1}$, queremos calcular el valor de

$$a_{ij}(\xi_1,\ldots,\xi_{n-1}) \qquad (1 \le i \le s \ , \ 0 \le j \le d)$$

Para ello, se considera el sistema lineal no homogéneo con coeficientes en k

$$\begin{cases} \sum_{j=0}^{d} \varepsilon_0^{\,j} z_{ij} = F_i(\xi_1,\ldots,\xi_{n-1},\varepsilon_0) \\ \vdots \\ \sum_{j=0}^{d} \varepsilon_d^{\,j} z_{ij} = F_i(\xi_1,\ldots,\xi_{n-1},\varepsilon_d) \end{cases}$$

Dado que la matriz asociada al sistema es la matriz A y que la única solución del sistema es $z_{ij} = a_{ij}(\xi_1,\ldots,\xi_{n-1})$, resulta que

$$\begin{pmatrix} a_{i0}(\xi_1,\ldots,\xi_{n-1}) \\ \vdots \\ a_{id}(\xi_1,\ldots,\xi_{n-1}) \end{pmatrix} = A^{-1} \cdot \begin{pmatrix} F_i(\xi_1,\ldots,\xi_{n-1},\varepsilon_0) \\ \vdots \\ F_i(\xi_1,\ldots,\xi_{n-1},\varepsilon_d) \end{pmatrix}$$

Luego, para hallar $a_{ij}(\xi_1,\ldots,\xi_{n-1})$ $(1 \le i \le s \ , \ 0 \le j \le d)$ necesitamos evaluar $F_1,\ldots,F_s$ en d+1 puntos y efectuar los productos de matrices. El costo de realizar esto es $\mathcal{L}.(d+1)+s.2(d+1)^2$.

Una vez obtenido el straight line program para a_{ij} $(1 \le i \le s, \ 0 \le j \le d)$ repetimos el procedimiento para la variable X_{n-1} (Notar que la matriz asociada a cada uno de los sistemas que van apareciendo es siempre la matriz A y, por lo tanto, el costo de calcularla e invertirla se cuenta sólo una vez).

Finalmente, después de iterar este procedimiento m veces y teniendo en cuenta que $m \le d$ y que $s \le \mathcal{L}$ se obtiene el straight line

program deseado de longitud $\mathcal{L}.d^{O(m)}$. ∎

3.2. Un bloque de cuantificadores existenciales sin desigualdades

Sean $X_1,\ldots,X_n$ indeterminadas sobre k y sea m un número natural tal que $1 \leq m \leq n$. Sean $F_1,\ldots,F_s \in k[X_1,\ldots,X_n]$ polinomios cuyos grados totales en las variables $X_{n-m+1},\ldots,X_n$ están acotados por un número natural $d \geq m$ y cuyos grados totales en las variables $X_1,\ldots,X_{n-m}$ están acotados por un número natural d'. Supondremos que los polinomios $F_1,\ldots,F_s \in k[X_1,\ldots,X_{n-m}][X_{n-m+1},\ldots,X_n]$ están codificados en forma densa en las variables $X_{n-m+1},\ldots,X_n$ y que sus coeficientes (que son elementos de $k[X_1,\ldots,X_{n-m}]$) están dados por un straight line program de longitud L. Sea $\mathcal{P} \subseteq \mathbb{A}^{n-m}(\bar{k})$ el conjunto definido por:

$$\mathcal{P} = \{ (x_1,\ldots,x_{n-m}) \in \bar{k}^{n-m} \;/\; \exists\, (x_{n-m+1},\ldots,x_n) \in \bar{k}^{m} :$$

$$F_1(x_1,\ldots,x_n) = 0 \wedge \ldots. \wedge F_s(x_1,\ldots,x_n) = 0 \}$$

Teorema 3.2.1. *A menos de una preparación previa, existe un algoritmo bien paralelizable y sin divisiones, de complejidad secuencial* $L + s^{O(1)}.d^{O(m)}$, *que permite describir al conjunto* $\mathcal{P}$ *de la siguiente manera:*

$$\mathcal{P} = \{ (x_1,\ldots,x_{n-m}) \in \bar{k}^{n-m} \;/\; \psi(x_1,\ldots,x_{n-m}) \}$$

donde ψ *es una fórmula libre de cuantificadores, es decir, una combinación booleana de fórmulas atómicas del tipo*

$$G_1(x_1,\ldots,x_{n-m}) = 0 \wedge \ldots. \wedge G_\lambda(x_1,\ldots,x_{n-m}) = 0 \wedge$$

$$\wedge G_{\lambda+1}(x_1,\ldots,x_{n-m}) \neq 0 \wedge \ldots. \wedge G_\mu(x_1,\ldots,x_{n-m}) \neq 0$$

con $G_1,\ldots,G_\mu \in k[X_1,\ldots,X_{n-m}]$.

La longitud de la fórmula ψ *es del orden de* $L + s^{O(1)}.d^{O(m)}$ *y en ella aparecen a lo sumo* $s^{O(1)}.d^{O(m)}$ *polinomios con grados acotados por* $d'.d^{O(m)}$, *dados por un straight line program de longitud* $L + s^{O(1)}.d^{O(m)}$.

Demostración: Fijemos un punto $(\xi_1,\ldots,\xi_{n-m}) \in \overline{k}^{n-m}$ y sea $\mathcal{K}$ el anillo $\mathcal{K} = k[\xi_1,\ldots,\xi_{n-m}]$.

Para cada $1 \leq i \leq s$ sea $f_i \in \mathcal{K}[X_{n-m+1},\ldots,X_n]$ el polinomio definido por $f_i = F_i(\xi_1,\ldots,\xi_{n-m},X_{n-m+1},\ldots,X_n)$.

Observemos que $(\xi_1,\ldots,\xi_{n-m}) \in \mathcal{P}$ si y sólo si la variedad

$$V = \{ (x_{n-m+1},\ldots,x_n) \in \overline{\mathcal{K}}^m \;/\; f_1(x_{n-m+1},\ldots,x_n) = 0 \wedge \ldots. \wedge$$

$$\wedge f_s(x_{n-m+1},\ldots,x_n) = 0 \} \subseteq \mathbb{A}^m(\overline{\mathcal{K}})$$

es no vacía.

Ahora bien, como $V \neq \emptyset$ es equivalente a $\dim V \geq 0$, nuestra intención es tratar de aplicar el algoritmo de [Gi-He, 1993] que calcula la dimensión de V (notar que la variedad V está definida por s polinomios en m variables a coeficientes en el anillo $\mathcal{K}$, dados en forma densa y cuyos grados están acotados por d). Este algoritmo se construye a partir de algoritmos básicos de álgebra lineal que utilizan straight line program para codificar polinomios y correct test sequences para realizar comparaciones. El problema

que surge al intentar aplicar el mencionado algoritmo en nuestro caso es la imposibilidad de decidir si un elemento de $\mathcal{K}$ es cero o no (notar que el punto fijado $(\xi_1,\ldots,\xi_{n-m})$ es un punto cualquiera de $\overline{k}^{n-m}$ y en general no es posible decidir si un polinomio a coeficientes en k evaluado en este punto es cero o no). Esto se solucionará modificando el algoritmo de manera conveniente.

Como en [Gi-He, 1993], introducimos $m^2+ m$ nuevas indeterminadas T_{rj} , T_r $(1 \le r,j \le m)$ y sea $\mathcal{R} = \mathcal{K}[T_{rj},T_r]_{1\le r,j\le m}$. En el caso en que la característica p de k sea positiva, en ciertas subrutinas del algoritmo de [Gi-He, 1993] aparece la necesidad de calcular raíces p-ésimas de elementos de $\mathcal{R}$. Como la cantidad de veces que debe realizarse dicha operación está acotada de antemano, el problema se solucionará mediante el cambio de las variables involucradas por potencias adecuadas de nuevas variables. Todo este proceso no incidirá en el resultado final ni en la complejidad total del algoritmo. Para mayores detalles, ver [Gi-He-Sa, 1993].

Para cada $1 \le r \le m$ sea $\lambda_r \in \mathcal{R}[X_{n-m+1},\ldots,X_n]$ la forma lineal definida por:

$$\lambda_r = T_{r1}\, X_{n-m+1} + \ldots. + T_{rm}\, X_n + T_r$$

Para cada $0 \le r \le m$ sea $W_r \subseteq \mathbb{A}^m(\overline{\mathcal{R}})$ la variedad definida por

$$W_r = \{\ (x_{n-m+1},\ldots,x_n) \in \overline{\mathcal{R}}^m \ /\ f_1(x_{n-m+1},\ldots,x_n) = 0 \wedge \ldots. \wedge$$

$$\wedge\ f_s(x_{n-m+1},\ldots,x_n) = 0 \wedge \qquad (1)$$

$$\wedge\ \lambda_1(x_{n-m+1},\ldots,x_n) = 0 \wedge \ldots. \wedge \lambda_r(x_{n-m+1},\ldots,x_n) = 0\ \}$$

Afirmación: $V = \emptyset$ si y sólo si para todo $0 \leq r \leq m$ $W_r = \emptyset$

En efecto, si $V = \emptyset$, por el Teorema de los ceros de Hilbert, existen polinomios $P_1,\ldots,P_s \in \mathcal{K}'[X_{n-m+1},\ldots,X_n]$ tales que

$$1 = \sum_{i=1}^{s} P_i . f_i$$

En consecuencia, $\forall\ 0 \leq r \leq m$,

$$1 = \sum_{i=1}^{s} P_i . f_i + \sum_{j=1}^{r} 0 \ .\lambda_j \qquad \text{en } \mathcal{R}'[X_{n-m+1},\ldots,X_n]$$

y, por lo tanto, $W_r = \emptyset$.

Por otra parte, como $V = W_0 \cap \overline{\mathcal{K}}^{\,m}$, si $W_0 = \emptyset$ resulta que $V = \emptyset$, lo que concluye la demostración de la afirmación.

Ahora encontraremos condiciones equivalentes a $V \neq \emptyset$ utilizando las variedades $W_0,\ldots,W_m$.

Sea $\Gamma = \{\gamma^{(1)},\ldots,\gamma^{(c)}\} \subseteq \mathcal{k}^m$ un conjunto de cardinal apropiado

$$c = s^{O(1)}.d^{O(m)} = (s+m)^{O(1)}.d^{O(m)}$$

como en [Gi-He, 1993], 3.4.7. Luego, para cada anillo efectivo $\mathcal{A}$ que contiene a $\mathcal{k}$ (donde la efectividad de $\mathcal{A}$ incluye la extracción de raíces p-ésimas en el caso en que la característica p de $\mathcal{k}$ sea positiva) y para cada variedad afin

$$W = \{\ (x_{n-m+1},\ldots,x_n) \in \overline{\mathcal{A}}^m \ /\ h_1(x_{n-m+1},\ldots,x_n) = 0 \wedge \ldots. \wedge$$

$$\wedge\ h_{s+m}(x_{n-m+1},\ldots,x_n) = 0\ \} \subseteq \mathbb{A}^m(\overline{\mathcal{A}})$$

definida por polinomios $h_1,\ldots,h_{s+m} \in \mathcal{A}[X_{n-m+1},\ldots,X_n]$, dados en forma densa y cuyos grados están acotados por d, se pueden calcu-

lar, utilizando el algoritmo bien paralelizable y sin divisiones de [Gi-He, 1993], de complejidad secuencial $(s + m)^{O(1)}.d^{O(m)} = s^{O(1)}.d^{O(m)}$, un elemento $0 \neq \alpha \in \mathcal{A}$, un elemento $\gamma = (\gamma_1,\ldots,\gamma_m)$ de Γ y, en consecuencia, un elemento $y \in k[X_{n-m+1},\ldots,X_n]$, $y = \gamma_1 X_{n-m+1} + \ldots + \gamma_m X_n$ y polinomios $r_1,\ldots,r_m \in \mathcal{A}[Z]$ de grado $d^{O(m)}$ en la indeterminada Z, tales que cada punto aislado $\omega \in W$ satisface

$$\alpha.\omega = (r_1(y(\omega)),\ldots,r_m(y(\omega)))$$

(para mayores detalles, ver [Gi-He, 1993], 3.4.7)

Esto es posible ya que las coordenadas de los elementos de Γ pueden ser elegidas en un subconjunto de $\mathcal{A}$ de cardinal apropiado, fijado de antemano y, por lo tanto, en k.

Además, todos los resultados intermedios de este algoritmo son polinomios, de grado $d^{O(m)}$ y evaluables en tiempo $s^{O(1)}\ d^{O(m)}$, en los coeficientes de $h_1,\ldots,h_{s+m}$, sobre k.

Es nuestra intención aplicar el algoritmo de [Gi-He, 1993] a cada una de las variedades W_i $(0 \leq i \leq m)$ definidas anteriormente, es decir, a variedades del tipo:

$$W = \{ (x_{n-m+1},\ldots,x_n) \in \overline{\mathcal{R}}^m \ / \ h_1(x_{n-m+1},\ldots,x_n) = 0 \wedge \qquad (2)$$
$$\wedge \ldots \wedge h_{s+m}(x_{n-m+1},\ldots,x_n) = 0 \ \} \subseteq \mathbb{A}^m(\overline{\mathcal{R}})$$

donde $h_1,\ldots,h_{s+m}$ son polinomios en $k[X_1,\ldots,X_n,T_{ij},T_i]_{1\leq i,j\leq m}$ de grado menor o igual que d en las variables $X_{n-m+1},\ldots,X_n$ y de grado menor o igual que d' en las variables $X_1,\ldots,X_{n-m},T_{ij},T_i$, dados en forma densa en las variables $X_{n-m+1},\ldots,X_n$ y en forma de straight line program de longitud L en las variables $X_1,\ldots,X_{n-m}$,

evaluados en el punto $(\xi_1,\ldots,\xi_{n-m})$ fijado al principio de la demostración.

En una primera parte, el algoritmo de [Gi-He, 1993] calcula, utilizando técnicas del álgebra lineal efectiva (cálculo de bases monomiales, matrices de transformaciones lineales en tales bases, bases standard, etc.), un polinomio $g \in \mathcal{R}[Y_1,\ldots,Y_m]$ y polinomios $g_1,\ldots,g_m \in \mathcal{R}[Y_1,\ldots,Y_m,Z]$, donde $Y_1,\ldots,Y_m,Z$ son nuevas indeterminadas sobre $\mathcal{R}$, que luego serán utilizados para encontrar el elemento α y los polinomios $r_1,\ldots,r_m$ buscados. Como el algoritmo mencionado realiza comparaciones entre los elementos del anillo de base y en nuestro caso no podemos determinar si un elemento $\beta \in \mathcal{R}$ es cero o no (ya que el punto $(\xi_1,\ldots,\xi_{n-m})$ fijado de antemano es cualquiera) cada vez que necesitemos decidir si $\beta \in \mathcal{R}$ es nulo o no, se considerarán las dos posibilidades: $\beta = 0$, $\beta \neq 0$. Para cada una de ellas se continuará con el algoritmo hasta obtener los polinomios $g, g_1,\ldots,g_m$ lo que producirá ramificaciones (selectores asociados a fórmulas) B_j $(1 \leq j \leq b)$, donde $b \leq s^{O(1)}\, d^{O(m)}$, del tipo:

$$B_j : \bigwedge_{i \in M} \beta_{ij} = 0 \ \wedge \ \bigwedge_{i \in N} \beta_{ij} \neq 0$$

donde $\# M + \# N \leq s^{O(1)}\, d^{O(m)}$ y cada $\beta_{ij} \in \mathcal{R}$ es un polinomio evaluable en tiempo secuencial $s^{O(1)}\, d^{O(m)}$ y de grado acotado por $d^{O(m)}$, en los coeficientes de $h_1,\ldots,h_{s+m}$.

En esta forma, para cada una de las ramificaciones obtenemos ciertos polinomios $g, g_1.,\ldots,g_m$ que dependen de ella.

Luego, en una segunda parte, el algoritmo de [Gi-He, 1993] en-

cuentra el elemento α y los polinomios $r_1,\ldots,r_m \in \mathcal{R}[Z]$ buscados utilizando los elementos del conjunto $\Gamma = \{\gamma^{(1)},\ldots,\gamma^{(c)}\} \subseteq k^m$ de la siguiente manera:

Primero calcula $g(\gamma^{(1)})$. Si $g(\gamma^{(1)}) \neq 0$ el algoritmo produce la salida $\alpha = g(\gamma^{(1)})$, $r_1(Z) = g_1(\gamma^{(1)},Z)$, ... , $r_m(Z) = g_m(\gamma^{(1)},Z)$. Si $g(\gamma^{(1)}) = 0$, calcula $g(\gamma^{(2)})$. Si $g(\gamma^{(2)}) \neq 0$ produce la salida $\alpha = g(\gamma^{(2)})$, $r_1(Z) = g_1(\gamma^{(2)},Z)$, ... , $r_m(Z) = g_m(\gamma^{(2)},Z)$. Si $g(\gamma^{(2)}) = 0$, el algoritmo continúa en forma análoga.

Para cada una de las ramificaciones obtenidas anteriormente continuamos con el algoritmo utilizando los polinomios $g,g_1,\ldots,g_m$ correspondientes a ella hasta obtener la salida. Como en nuestro caso no es posible determinar si $g(\gamma^{(1)})$ es cero para $\gamma^{(1)} \in \Gamma$, tomando para cada $1 \leq i \leq c$

$$\alpha^{\gamma^{(1)}} = g(\gamma^{(1)}),\ r_1^{\gamma^{(1)}}(Z) = g_1(\gamma^{(1)},Z),\ldots,r_m^{\gamma^{(1)}}(Z) = g_m(\gamma^{(1)},Z)$$

al considerar todas las posibilidades cada condición B_j se ramifica en la forma:

B_j

$\alpha_j^{\gamma^{(1)}} \neq 0$

$\alpha_j^{\gamma^{(1)}} = 0 \wedge \alpha_j^{\gamma^{(2)}} \neq 0$

$\alpha_j^{\gamma^{(1)}} = 0 \wedge \alpha_j^{\gamma^{(2)}} = 0 \wedge \alpha_j^{\gamma^{(3)}} \neq 0$

.....

.....

$\bigwedge_{i=1}^{c-1} \alpha_j^{\gamma^{(1)}} = 0 \wedge \alpha_j^{\gamma^{(c)}} \neq 0$

De esta manera obtenemos un nuevo algoritmo que contiene rami-

ficaciones $B_j^{(r)}$ ($1 \le r \le c$, $1 \le j \le b$), donde $c = \#\,\Gamma \le s^{O(1)}\, d^{O(m)}$ y $b \le s^{O(1)}\, d^{O(m)}$, del tipo:

$$B_j^{(r)} : \bigwedge_{i \in M} \beta_{ij} = 0 \;\wedge\; \bigwedge_{i \in N} \beta_{ij} \neq 0 \;\wedge\; \bigwedge_{i=1}^{r-1} \alpha_j^{\gamma^{(i)}} = 0 \;\wedge\; \alpha_j^{\gamma^{(r)}} \neq 0$$

donde $\#\,M + \#\,N \le s^{O(1)}\, d^{O(m)}$ y cada $\beta_{ij} \in \mathcal{R}$ y cada $\alpha_j^{\gamma^{(i)}} \in \mathcal{R}$ es un polinomio evaluable en tiempo secuencial $s^{O(1)}\, d^{O(m)}$ y de grado acotado por $d^{O(m)}$, en los coeficientes de $h_1, \ldots, h_{s+m}$. Para cada una de las ramificaciones $B_j^{(r)}$ este nuevo algoritmo produce la salida $\alpha = \alpha_j^{\gamma^{(r)}}$, $r_1 = r_1^{\gamma^{(r)}}, \ldots, r_m = r_m^{\gamma^{(r)}}$. Notemos que tanto $\alpha_j^{\gamma^{(r)}}$ como los coeficientes de $r_1^{\gamma^{(r)}}, \ldots, r_m^{\gamma^{(r)}}$ son polinomios a coeficientes en k en las indeterminadas $X_1, \ldots, X_{n-m}, T_{ij}, T_i$ ($1 \le i, j \le m$), de grado acotado por $d' . d^{O(m)}$ evaluados en el punto $(\xi_1, \ldots, \xi_{n-m})$.

Incluyendo las ramificaciones en la salida conseguimos un nuevo algoritmo que, cuando lo apliquemos a una variedad W como en (2), producirá una salida del tipo:

$$\left\{ B_j^{(r)} \;;\; \alpha_j^{\gamma^{(r)}} \;;\; r_1^{\gamma^{(r)}}, \ldots, r_m^{\gamma^{(r)}} \right\}_{\substack{1 \le j \le b \\ 1 \le r \le c}}$$

Para el punto $(\xi_1, \ldots, \xi_{n-m})$ ya fijado existen únicos j_0 y r_0 con $1 \le j_0 \le b$ y $1 \le r_0 \le c$ tales que $(\xi_1, \ldots, \xi_{n-m})$ satisface $B_{j_0}^{(r_0)}$ (la existencia está garantizada por el refinamiento del lema del elemento primitivo ([Gi-He, 1993], 3.4.7.) y la unicidad se desprende de la definición de $B_j^{(r)}$), el $\alpha_{j_0}^{\gamma^{(r_0)}}$ correspondiente es distinto de cero y además, todo punto aislado $\omega \in W$ satisface:

$$\alpha_{j_0}^{\gamma^{(r_0)}} \cdot \omega = \left(r_1^{\gamma^{(r_0)}}(y(\omega)), \ldots, r_m^{\gamma^{(r_0)}}(y(\omega)) \right)$$

Ante la imposibilidad de decidir cuáles son el j_0 y el r_0 que corresponden al punto fijado, se continúa el algoritmo para cada j y cada r ($1 \le j \le b$ y $1 \le r \le c$) de la siguiente manera:

Para cada h_i ($1 \le i \le s+m$) que aparece en la definición de W sea

$$P_i^{j,r} = \left(\alpha_j^{\gamma^{(r)}}\right)^d \cdot h_i\left(\frac{r_1^{\gamma^{(r)}}}{\alpha_j^{\gamma^{(r)}}}, \ldots, \frac{r_m^{\gamma^{(r)}}}{\alpha_j^{\gamma^{(r)}}} \right) \in \mathcal{R}[Z]$$

En el caso en que $j = j_0$ y $r = r_0$ estos polinomios $P_i^{j,r}$ poseen las propiedades siguientes:

⊙ Si W contiene puntos aislados, entonces $P_1^{j,r}, \ldots, P_{s+m}^{j,r}$ no son coprimos (como polinomios a coeficientes en $\mathcal{R}'$ en la variable Z)

⊙⊙ Si $P_1^{j,r}, \ldots, P_{s+m}^{j,r}$ no son coprimos en $\mathcal{R}'[Z]$ entonces $W \neq \emptyset$

En efecto, como $\alpha_j^{\gamma^{(r)}} \neq 0$ el primer punto se deduce del hecho de que todo punto aislado $\omega \in W$ satisface

$$\alpha_j^{\gamma^{(r)}} \cdot \omega = \left(r_1^{\gamma^{(r)}}(y(\omega)), \ldots, r_m^{\gamma^{(r)}}(y(\omega)) \right)$$

y el segundo punto es trivial.

Cuando además W contiene a lo sumo puntos aislados, de ⊙ y ⊙⊙ resulta claro que la condición $W = \emptyset$ es equivalente a que los polinomios $P_1^{j,r}, \ldots, P_{s+m}^{j,r}$ sean coprimos en $\mathcal{R}'[Z]$.

De esta manera, cuando continuamos con el algoritmo para todo

j,r, la condición W = ø resulta equivalente a

$$\bigvee_{\substack{1\le j\le b\\ 1\le r\le c}} \left(B_j^{(r)} \wedge P_1^{j,r},\ldots,P_{s+m}^{j,r} \text{ sean coprimos en } \mathcal{R}'[Z] \right)$$

Ahora bien, dados j y r, los polinomios $P_1^{j,r},\ldots,P_{s+m}^{j,r}$ son coprimos en $\mathcal{R}'[Z]$ si y sólo si existen $Q_1^{j,r},\ldots,Q_{s+m}^{j,r} \in \mathcal{R}'[Z]$, de grado en Z acotado por $d^{O(m)}$, tales que

$$1 = \sum_{i=1}^{s+m} P_i^{j,r} \cdot Q_i^{j,r} \qquad (3)$$

si y sólo si el sistema lineal no homogéneo con coeficientes en $\mathcal{R}$ determinado por (3) tiene una solución en $\mathcal{R}'$, lo que es equivalente a que el rango de la matriz del sistema homogéneo asociado sea igual al rango de la matriz ampliada. Para obtener los coeficientes de la matriz del sistema (que vendrán dados por medio de un straight line program) necesitamos tener a $P_1^{j,r},\ldots,P_{s+m}^{j,r}$ escritos en forma densa en la variable Z. Para ello, primero eliminaremos las divisiones por $\alpha_j^{\gamma^{(r)}}$, obteniendo así un staight line program para $P_1^{j,r},\ldots,P_{s+m}^{j,r}$.

Como los polinomios h_i ($1\le i\le s+m$) ya vienen dados en forma densa en las variables $X_{n-m+1},\ldots,X_n$, bastará introducir una nueva variable Y, hacer los $h_i(X_{n-m+1},\ldots,X_n)$ ($1\le i\le s+m$) homogéneos de grado d utilizando la variable Y y, si $\tilde{h}_i(X_{n-m+1},\ldots,X_n,Y)$ ($1\le i\le s+m$) son los nuevos polinomios obtenidos, entonces

$$P_i^{j,r} = \tilde{h}_i \left(r_1^{\gamma^{(r)}}, \ldots, r_m^{\gamma^{(r)}}, \alpha_j^{\gamma^{(r)}} \right)$$

Ahora, escribir estos polinomios en forma densa en la variable Z puede hacerse interpolando (ver Proposición 3.1.1.), teniendo en

cuenta que el grado de $P_1^{j,r},\ldots,P_{s+m}^{j,r}$ en dicha variable está acotado por $d^{O(m)}$.

El sistema consta de $d^{O(m)}$ ecuaciones y $s^{O(1)}\ d^{O(m)}$ incógnitas y, siendo los coeficientes de la matriz del sistema precisamente los coeficientes de $P_1^{j,r},\ldots,P_{s+m}^{j,r}$, resulta que éstos son polinomios en $k[X_1,\ldots,X_{n-m},T_{1j},T_1]_{1\le l,\,j\le m}$ de grado acotado por $d'.d^{O(m)}$, dados por un straight line program de longitud $L + s^{O(1)}\ d^{O(m)}$, evaluados en $(\xi_1,\ldots,\xi_{n-m})$.

Sea B la matriz que se obtiene agregándole una columna de ceros a la matriz del sistema homogéneo asociado y sea $\bar{B}$ la matriz ampliada del sistema. Utilizando las técnicas de Berkowitz-Mulmuley ([Ber, 1984] y [Mul, 1986]) introducimos dos nuevas indeterminadas Z_1 y Z_2 y calculamos los polinomios característicos de las matrices cuadradas A y $\bar{A}$ que se obtienen a partir de B y $\bar{B}$ (notar que ambos característicos tienen igual grado pues las matrices tienen las mismas dimensiones):

$$\chi_A = G_0^{j,r} + G_1^{j,r}\ \lambda + \ldots. + G_t^{j,r}\ \lambda^t + \lambda^{t+1}$$
$$\chi_{\bar{A}} = H_0^{j,r} + H_1^{j,r}\ \lambda + \ldots. + H_t^{j,r}\ \lambda^t + \lambda^{t+1}$$

donde $t \le d^{O(m)}$ y $G_0^{j,r}, G_1^{j,r},\ldots,G_t^{j,r}$, $H_0^{j,r}$, $H_1^{j,r},\ldots,$ $H_t^{j,r}$ son polinomios en $k[X_1,\ldots,X_{n-m},T_{1k},T_1,Z_1,Z_2]_{1\le l,\,k\le m}$, de grado acotado por $d'.d^{O(m)}$, dados por un straight line program de longitud $L + s^{O(1)}\ d^{O(m)}$, evaluados en el punto $(\xi_1,\ldots,\xi_{n-m})$.

Por lo tanto, $P_1^{j,r},\ldots,P_{s+m}^{j,r}$ son coprimos en $\mathcal{R}'[Z]$ si y sólo si las multiplicidades de cero como raíz en ambos característicos

coinciden, lo que se traduce en la condición:

$$D_j^{(r)} : (G_0^{j,r} \neq 0 \wedge H_0^{j,r} \neq 0) \vee (G_0^{j,r} = 0 \wedge H_0^{j,r} = 0 \wedge G_1^{j,r} \neq 0 \wedge$$

$$\wedge H_1^{j,r} \neq 0) \vee \vee (G_0^{j,r} = 0 \wedge H_0^{j,r} = 0 \wedge \ldots \wedge G_{t-1}^{j,r} = 0 \wedge$$

$$\wedge H_{t-1}^{j,r} = 0 \wedge G_t^{j,r} \neq 0 \wedge H_t^{j,r} \neq 0)$$

En resumen, si W es como en (2) y contiene a lo sumo puntos aislados entonces

$$W = \emptyset \Leftrightarrow \bigvee_{\substack{1 \leq j \leq b \\ 1 \leq r \leq c}} \left(B_j^{(r)} \wedge D_j^{(r)} \right)$$

Observemos que en las condiciones $B_j^{(r)}$ y $D_j^{(r)}$ aparecen polinomios en $k[\xi_1,\ldots,\xi_{n-m},T_{ik},T_i,Z_1,Z_2]_{1\leq i,k\leq m}$ y queremos tener condiciones equivalentes a ellas pero que sólo involucren elementos de $k[\xi_1,\ldots,\xi_{n-m}]$. Como los polinomios que aparecen en dichas condiciones son de grado acotado por $D \leq d^{O(m)}$ en las $m^2 + m + 2$ variables T_{ik}, T_i, Z_1, Z_2 ($1 \leq i,k \leq m$) y están dados por un straight line program de longitud $v = s^{O(1)}\, d^{O(m)}$ en dichas variables dado que los polinomios de entrada $F_1,\ldots,F_s$ vienen dados en forma densa en las indeterminadas $X_{n-m+1},\ldots,X_n$, utilizaremos el resultado de [He-Sch, 1982] para obtener las condiciones deseadas:

Sea $\Delta \subseteq k^{m^2+m+2}$ un conjunto de cardinal $6(v+m^2+m+2)(v+m^2+m+3) \leq$ $\leq s^{O(1)}\, d^{O(m)}$ que satisface:

$\forall\ P \in k[\xi_1, \ldots,\xi_{n-m}][T_{ik},T_i,Z_1,Z_2]_{1\leq i,k\leq m}$ de grado acotado por D y evaluable por medio de un straight line program de longitud acotada por v, $P = 0 \Leftrightarrow P(\delta) = 0 \quad \forall\ \delta \in \Delta$

De esta forma, cada condición

$$P(\xi_1,\ldots,\xi_{n-m},T_{ik},T_i,Z_1,Z_2)_{1\le i,k\le m} = 0$$

se traduce en

$$\bigwedge_{\delta\in\Delta} P(\xi_1,\ldots,\xi_{n-m},\delta) = 0$$

Análogamente, cada condición

$$P(\xi_1,\ldots,\xi_{n-m},T_{ik},T_i,Z_1,Z_2)_{1\le i,k\le m} \neq 0$$

se traduce en

$$\bigvee_{\delta\in\Delta} P(\xi_1,\ldots,\xi_{n-m},\delta) \neq 0$$

Cuando aplicamos a una variedad W del tipo (2) el algoritmo de complejidad secuencial $L + s^{O(1)}\ d^{O(m)}$ que hemos construído, obtenemos como salida una fórmula ψ_W libre de cuantificadores, que es una combinación booleana de fórmulas atómicas del tipo:

$$g_1(\xi_1,\ldots,\xi_{n-m}) = 0 \wedge \ldots\ldots \wedge g_{h'}(\xi_1,\ldots,\xi_{n-m}) = 0 \wedge$$

$$\wedge\ g_{h'+1}(\xi_1,\ldots,\xi_{n-m}) \neq 0 \wedge \ldots\ldots \wedge g_h(\xi_1,\ldots,\xi_{n-m}) \neq 0$$

tal que

a) $|\psi_W| \le s^{O(1)}\ d^{O(m)}$

b) cada g_i es un polinomio en $k[X_1, \ldots, X_{n-m}]$ de grado acotado por $d'.\ d^{O(m)}$

c) los polinomios g_i vienen dados por medio de un straight line program de longitud $L + s^{O(1)}.d^{O(m)}$.

Además, si W contiene a lo sumo puntos aislados, entonces:

$$W = \emptyset \text{ si y sólo si } \psi_W$$

Para cada una de las variedades W_r $(0 \le r \le m)$ que definimos en

(1) aplicamos el algoritmo anterior obteniendo como salida las fórmulas $\psi_{W_0}, \ldots, \psi_{W_m}$.

Afirmación: $V = \emptyset$ si y sólo si $\bigwedge_{r=0}^{m} \psi_{W_r}$

En efecto, si $V = \emptyset$, entonces $W_r = \emptyset \quad \forall\ 0 \leq r \leq m$. En particular, cada W_r contiene a lo sumo puntos aislados, de donde ψ_{W_r} para todo $0 \leq r \leq m$.

Por otra parte, supongamos $\bigwedge_{r=0}^{m} \psi_{W_r}$, entonces ψ_{W_m}. Como por construcción W_m tiene a lo sumo puntos aislados, entonces $W_m = \emptyset$ y, en consecuencia, W_{m-1} tiene a lo sumo puntos aislados. Luego $W_{m-1} = \emptyset$ pues $\psi_{W_{m-1}}$. Repitiendo este razonamiento, resulta que $W_r = \emptyset$ para todo $0 \leq r \leq m$ y por lo tanto $V = \emptyset$.

Luego,

$$V \neq \emptyset \text{ sii } \bigvee_{r=0}^{m} \neg\, \psi_{W_r} = \psi(\xi_1, \ldots, \xi_{n-m})$$

Como se desprende de la construcción del algoritmo, los polinomios que aparecen en la fórmula ψ hallada no dependen del punto $(\xi_1, \ldots, \xi_{n-m})$ fijado: hemos obtenido una fórmula $\psi(X_1, \ldots, X_{n-m})$ libre de cuantificadores que satisface

$$\{ \ (x_1, \ldots, x_{n-m}) \in \overline{k}^{n-m} \ / \ \exists\ (x_{n-m+1}, \ldots, x_n) \in \overline{k}^{m} \ :$$

$$F_1(x_1, \ldots, x_n) = 0 \wedge \ldots. \wedge F_s(x_1, \ldots, x_n) = 0 \ \} =$$

$$= \{ \ (x_1, \ldots, x_{n-m}) \in \overline{k}^{n-m} \ / \ \psi(x_1, \ldots, x_{n-m}) \ \} \qquad \blacksquare$$

Observación 3.2.2. En el caso en que los polinomios $F_1,\ldots,F_s$ estén codificados por un straight line program de longitud $\mathcal{L}$ en todas las variables, se procederá primero a escribirlos en forma densa en las variables $X_{n-m+1},\ldots,X_n$ con sus coeficientes (que son elementos de $k[X_1,\ldots,X_{n-m}]$) codificados en forma de straight line program (ver Proposición 3.1.1.) y se aplicará luego el algoritmo del Teorema 3.2.1., para $L = \mathcal{L} \cdot d^{O(m)}$.

Notar además que, si los polinomios $F_1,\ldots,F_s$ están codificados en forma densa en todas las variables, se los puede escribir de manera obvia en forma densa en las variables $X_{n-m+1},\ldots,X_n$ dando sus coeficientes (que son elementos de $k[X_1,\ldots,X_{n-m}]$) por medio de un straight line program de longitud $s.d'^{O(n-m)}.d^{O(m)}$ y se aplicará luego el algoritmo del Teorema 3.2.1.

3.3. Ejemplo

Sean $d,r \in \mathbb{N}$ tales que $d \geq r$ y sea φ la fórmula:

$$\exists x_1 \exists x_2 \ldots \exists x_{r-1} : x_1^d.y_1 - 1 = 0 \wedge x_2^d.y_2 - x_1 = 0 \wedge$$

$$\wedge x_3^d.y_3 - x_2 = 0 \wedge \ldots. \wedge x_{r-1}^d.y_{r-1} - x_{r-2} = 0 \wedge$$

$$\wedge y_r^d - x_{r-1} = 0$$

En este caso hay r variables libres y r-1 variables ligadas, aparecen r polinomios cuyos grados en las variables libres están acotados por d y cuyos grados en las variables ligadas también están acotados por d. Estos polinomios pueden ser codificados por

medio de un straight line program de longitud $L = 4r-1+2r[\log_2 d]$, donde $[\log_2 d]$ es la parte entera del logaritmo en base 2 de d.

Aplicando el algoritmo descripto en el Teorema 3.2.1. se obtiene una fórmula libre de cuantificadores equivalente a φ y el tiempo requerido para esto es del orden de $d^{O(r)}$.

Siguiendo las ideas de [Fi-Ga-Mo, 1990], analizaremos qué pasa en este caso en el modelo de la representación densa de polinomios.

Es evidente que φ es equivalente a la siguiente fórmula libre de cuantificadores:

$$Y_r^{d^r} \cdot Y_{r-1}^{d^{r-2}} \cdot Y_{r-2}^{d^{r-3}} \cdots Y_2^{d} \cdot Y_1 - 1 = 0$$

Sea ψ una fórmula libre de cuantificadores equivalente a φ.

Luego, $\{(Y_1,\ldots,Y_r) \in \overline{k}^r / \ \psi(Y_1,\ldots,Y_r)\}$ debe ser la variedad algebraica de dimensión r-1

$$V = \{(Y_1,\ldots,Y_r) \in \overline{k}^r / \ Y_r^{d^r} \cdot Y_{r-1}^{d^{r-2}} \cdot Y_{r-2}^{d^{r-3}} \cdots Y_2^{d} \cdot Y_1 - 1 = 0\}$$

que es una variedad irreducible pues el polinomio

$$Y_r^{d^r} \cdot Y_{r-1}^{d^{r-2}} \cdot Y_{r-2}^{d^{r-3}} \cdots Y_2^{d} \cdot Y_1 - 1$$

es irreducible en $k[Y_1,\ldots,Y_r]$.

Sean $G_1,\ldots,G_t$ $k[Y_1,\ldots,Y_r]$los polinomios que aparecen en la fórmula ψ. El hecho conocido de que toda fórmula sin cuantificadores puede escribirse en la forma normal disyuntiva nos permite describir a la variedad V de la siguiente manera:

$$V = \bigcup \{(y_1,\ldots,y_r) \in \overline{k}^r / \; G_{i_1}(y_1,\ldots,y_r) = 0 \wedge$$

$$\wedge\, G_{i_2}(y_1,\ldots,y_r) = 0 \wedge \ldots \wedge G_{i_k}(y_1,\ldots,y_r) = 0 \wedge$$

$$\wedge\, G_{i_{k+1}}(y_1,\ldots,y_r) \neq 0 \wedge \ldots \wedge G_{i_t}(y_1,\ldots,y_r) \neq 0\}$$

y como V es cerrado entonces

$$V = \bigcup \{(y_1,\ldots,y_r) \in \overline{k}^r / \; G_{i_1}(y_1,\ldots,y_r) = 0 \wedge$$

$$\overline{\wedge\, G_{i_2}(y_1,\ldots,y_r) = 0 \wedge \ldots \wedge G_{i_k}(y_1,\ldots,y_r) = 0 \wedge}$$

$$\overline{\wedge\, G_{i_{k+1}}(y_1,\ldots,y_r) \neq 0 \wedge \ldots \wedge G_{i_t}(y_1,\ldots,y_r) \neq 0\}}$$

Siendo V irreducible, resulta que V debe ser uno de los conjuntos que aparecen en esta última unión y como dim V=r-1 y V es cerrado , este conjunto no puede ser de la forma:

$$\overline{\{(y_1,\ldots,y_r) \in \overline{k}^r / \; G_1(y_1,\ldots,y_r) \neq 0 \wedge \ldots \wedge G_t(y_1,\ldots,y_r) \neq 0\}}$$

Luego, existe i, $1 \leq i \leq t$, tal que

$$V \subseteq \{(y_1,\ldots,y_r) \in \overline{k}^r / \; G_i(y_1,\ldots,y_r) = 0\}$$

de donde $Y_r^{d^r} . Y_{r-1}^{d^{r-2}} . Y_{r-2}^{d^{r-3}} \ldots . Y_2^{d} . Y_1 - 1$ divide a G_{i_1}.

Esto muestra que cualquier fórmula libre de cuantificadores equivalente a φ necesariamente contiene un polinomio en r variables a coeficientes en k de grado mayor o igual que d^r y por lo tanto cualquier algoritmo en el modelo de la representación densa de polinomios tendrá una complejidad no inferior a d^{r^2}.

3.4. Un bloque de cuantificadores existenciales con desigualdades

Como antes, sean $X_1,\ldots,X_n$ indeterminadas sobre k y sea m un número natural tal que $1 \leq m \leq n$. Sean $F_1,\ldots,F_s \in k[X_1,\ldots,X_n]$ polinomios cuyos grados en las variables $X_{n-m+1},\ldots,X_n$ están acotados por un número natural $d \geq m$ y cuyos grados en las variables $X_1,\ldots,X_{n-m}$ están acotados por un número natural d'.

Sean además $G_1,\ldots,G_{s'} \in k[X_1,\ldots,X_n]$ polinomios cuyos grados en las variables $X_{n-m+1},\ldots,X_n$ están acotados por un número natural δ y cuyos grados en las variables $X_1,\ldots,X_{n-m}$ están acotados por un número natural δ'.

Supondremos que los polinomios $F_1,\ldots,F_s,G_1,\ldots,G_{s'}$ están dados por un straight line program de longitud L.

Sea $\mathcal{P} \subseteq \mathbb{A}^{n-m}(\overline{k})$ el conjunto definido por:

$$\mathcal{P} = \{ (x_1,\ldots,x_{n-m}) \in \overline{k}^{n-m} \;/\; \exists\, (x_{n-m+1},\ldots,x_n) \in \overline{k}^{m} :$$

$$F_1(x_1,\ldots,x_n) = 0 \wedge \ldots. \wedge F_s(x_1,\ldots,x_n) = 0 \wedge$$

$$\wedge\, G_1(x_1,\ldots,x_n) \neq 0 \wedge \ldots. \wedge G_{s'}(x_1,\ldots,x_n) \neq 0 \}$$

Utilizando el truco de Rabinowitz, si Y es una nueva indeterminada y $G = 1 - Y \cdot \prod_{i=1}^{s'} G_i$, $\mathcal{P}$ se puede describir de la siguiente manera:

$$\mathcal{P} = \{ (x_1,\ldots,x_{n-m}) \in \overline{k}^{n-m} \;/\; \exists\, (x_{n-m+1},\ldots,x_n,y) \in \overline{k}^{m+1} :$$

$$F_1(x_1,\ldots,x_n) = 0 \wedge \ldots. \wedge F_s(x_1,\ldots,x_n) = 0 \wedge$$

$$\wedge\, G(x_1,\ldots,x_n,y) = 0 \}$$

Notar que éstos son s+1 polinomios dados por un straight line program de longitud L + s' + 1, cuyos grados en las indeterminadas $X_{n-m+1},\ldots,X_n,Y$ están acotados por D = máx {d,s'δ+1} y cuyos grados en $X_1,\ldots,X_{n-m}$ están acotados por D'= max {d', s'δ'},

Teniendo en cuenta la Observación 3.2.2., el algoritmo del Teorema 3.2.1., de complejidad secuencial $(L+s').D^{O(m)}+ s^{O(1)}.D^{O(m)}$, permite describir al conjunto $\mathcal{P}$ mediante una fórmula ψ libre de cuantificadores de longitud $(L + s').D^{O(m)} + s^{O(1)}.D^{O(m)}$. Los polinomios que aparecen en ψ vienen dados por un straight line program de longitud $(L + s').D^{O(m)} + s^{O(1)}.D^{O(m)}$ y sus grados son del orden de $D'.D^{O(m)}$.

Como puede verse, las cotas obtenidas de esta manera dependen polinomialmente de $s'^{O(m)}$ y de $\delta^{O(m)}$.

Sin embargo, esta dependencia no es intrínseca del problema sino del algoritmo utilizado tal como lo muestra el siguiente:

Teorema 3.4.1. *A menos de una preparación previa, existe un algoritmo bien paralelizable y sin divisiones de complejidad secuencial acotada por* $L^2\ (s.s'.\delta)^{O(1)}\ d^{O(m)}$, *que permite describir al conjunto* $\mathcal{P}$ *de la siguiente manera:*

$$\mathcal{P} = \{\ (x_1,\ldots,x_{n-m}) \in \overline{k}^{n-m}\ /\ \psi(x_1,\ldots,x_{n-m})\ \}$$

donde ψ *es una fórmula libre de cuantificadores, es decir, una combinación booleana de fórmulas atómicas del tipo*

$$H_1(x_1,\ldots,x_{n-m}) = 0 \wedge \ldots. \wedge H_\lambda(x_1,\ldots,x_{n-m}) = 0 \wedge$$

$$\wedge\ H_{\lambda+1}(x_1,\ldots,x_{n-m}) \neq 0 \wedge \ldots. \wedge H_\mu(x_1,\ldots,x_{n-m}) \neq 0$$

con $H_1,\ldots,H_\mu \in k[X_1,\ldots,X_{n-m}]$.

La longitud de la fórmula ψ*, así como también la cantidad de polinomios que aparecen en ella, es del orden de* $L^2\ (s.s'.\delta)^{O(1)}\ d^{O(m)}$. *Además, los polinomios de salida vendrán dados por un straight line program de longitud* $L^2\ (s.s'.\delta)^{O(1)}\ d^{O(m)}$ *y sus grados serán del orden de* $\delta' d'(s'.\delta)^{O(1)}\ d^{O(m)}$.

Demostración: La idea de la demostración es modificar convenientemente el algoritmo del Teorema 3.2.1. Primero escribimos los polinomios $F_1,\ldots,\ F_s$ en forma densa en las variables $X_{n-m+1},\ldots,X_n$ con sus coeficientes en $k[X_1,\ldots,X_{n-m}]$ codificados por medio de un straight line program. (Ver Proposición 3.1.1.)

Ahora, fijemos como antes un punto $(\xi_1,\ldots,\xi_{n-m}) \in \overline{k}^{n-m}$ y sea $\mathcal{K} = = k[\xi_1,\ldots,\xi_{n-m}]$.

Para cada $1 \le i \le s$ sea $f_i \in \mathcal{K}[X_{n-m+1},\ldots,X_n]$ el polinomio definido por

$$f_i = F_i(\xi_1,\ldots,\xi_{n-m},X_{n-m+1},\ldots,X_n)$$

y, para cada $1 \le j \le s'$, sea $g_j \in \mathcal{K}[X_{n-m+1},\ldots,X_n]$ el polinomio definido por

$$g_j = G_i(\xi_1,\ldots,\xi_{n-m},X_{n-m+1},\ldots,X_n)$$

Sea $V \subseteq \mathbb{A}^m(\overline{\mathcal{K}})$ la variedad

$$V = \{\ (x_{n-m+1},\ldots,x_n) \in \overline{\mathcal{K}}^m \ /\ f_1(x_{n-m+1},\ldots,x_n) = 0 \wedge \ldots. \wedge$$

$$\wedge\ f_s(x_{n-m+1},\ldots,x_n) = 0\ \}$$

y sea $\mathcal{U} \subseteq \mathbb{A}^m(\overline{\mathcal{K}})$ el abierto

$$\mathcal{U} = \{ (x_{n-m+1},\ldots,x_n) \in \overline{K}^m \;/\; g_1(x_{n-m+1},\ldots,x_n) \neq 0 \wedge \ldots \wedge g_{s'}(x_{n-m+1},\ldots,x_n) \neq 0 \}$$

Como antes, introducimos nuevas indeterminadas T_{rj}, T_r $(1 \leq r,j \leq m)$.

Sea $\mathcal{R} = K[T_{rj}, T_r]_{1\leq r,j\leq m}$. Para cada $1 \leq r \leq m$ definimos la forma lineal $\lambda_r \in \mathcal{R}[X_{n-m+1},\ldots,X_n]$ en la forma:

$$\lambda_r = T_{r1} X_{n-m+1} + \ldots + T_{rm} X_n + T_r$$

Para cada $0 \leq r \leq m$ sea $W_r \subseteq \mathbb{A}^m(\overline{\mathcal{R}})$ la variedad

$$W_r = \{ (x_{n-m+1},\ldots,x_n) \in \overline{\mathcal{R}}^m \;/\; f_1(x_{n-m+1},\ldots,x_n) = 0 \wedge \ldots \wedge f_s(x_{n-m+1},\ldots,x_n) = 0 \wedge \lambda_1(x_{n-m+1},\ldots,x_n) = 0 \wedge \ldots \wedge \lambda_r(x_{n-m+1},\ldots,x_n) = 0 \} \quad (4)$$

y sea $\mathcal{U}' \subseteq \mathbb{A}^m(\mathcal{R})$ el abierto

$$\mathcal{U}' = \{ (x_{n-m+1},\ldots,x_n) \in \overline{\mathcal{R}}^m \;/\; g_1(x_{n-m+1},\ldots,x_n) \neq 0 \wedge \ldots \wedge g_{s'}(x_{n-m+1},\ldots,x_n) \neq 0 \}$$

Afirmación: $V \cap \mathcal{U} = \emptyset$ si y sólo si $W_r \cap \mathcal{U}' = \emptyset \;\; \forall\, 0 \leq r \leq m$

En efecto, si $W_r \cap \mathcal{U}' = \emptyset \;\; \forall\, 0 \leq r \leq m$ entonces, en particular, $W_0 \cap \mathcal{U}' = \emptyset$. Luego, $W_0 \cap \mathcal{U}' \cap \overline{K}^m = \emptyset$, de donde resulta que $V \cap \mathcal{U} = (W_0 \cap \overline{K}^m) \cap (\mathcal{U}' \cap \overline{K}^m) = \emptyset$.

Por otra parte, si $V \cap \mathcal{U} = \emptyset$, introduciendo una nueva indeterminada Y sobre $\overline{K}$ y utilizando el truco de Rabinowitz obtenemos que

$$\{ \ (x_{n-m+1},\ldots,x_n,y) \in \overline{K}^{m+1} \ / \ f_1(x_{n-m+1},\ldots,x_n) = 0 \wedge \ldots \wedge$$

$$\wedge f_s(x_{n-m+1},\ldots,x_n) = 0 \wedge 1 - y.\prod_{i=1}^{s'} g_i(x_{n-m+1},\ldots,x_n) = 0 \ \}$$

es vacío. Luego, por el Teorema de los ceros de Hilbert, existen $P_i \in K'[X_{n-m+1},\ldots,X_n,Y]$ $(1 \leq i \leq s+1)$ tales que

$$1 = \sum_{i=1}^{s} P_i.f_i + P_{s+1}.(1 - Y.\prod_{i=1}^{s'} g_i)$$

Es decir, para todo $0 \leq r \leq m$,

$$1 = \sum_{i=1}^{s} P_i.f_i + \sum_{j=1}^{r} 0.\lambda_j + P_{s+1}.(1 - Y.\prod_{i=1}^{s'} g_i)$$

en $K'[X_{n-m+1},\ldots,X_n,Y]$ y por lo tanto $W_r \cap U' = \emptyset$, $\forall \ 0 \leq r \leq m$ como habíamos afirmado.

Sean ahora Γ y c como en el Teorema 3.2.1. y sea W una variedad del tipo:

$$W = \{ \ (x_{n-m+1},\ldots,x_n) \in \overline{K}^m \ / \ h_1(x_{n-m+1},\ldots,x_n) = 0 \wedge \qquad (5)$$

$$\wedge \ldots \wedge h_{s+m}(x_{n-m+1},\ldots,x_n) = 0 \ \} \subseteq \mathbb{A}^m(\overline{K})$$

donde $h_1,\ldots,h_{s+m}$ son polinomios en $k[X_1,\ldots,X_n,T_{ij},T_i]_{1\leq i,j\leq m}$ de grado menor o igual que d_c en las variables $X_{n-m+1},\ldots,X_n$ y de grado menor o igual que d' en las variables $X_1,\ldots,X_{n-m},T_{ij},T_i$, dados en forma densa en las variables $X_{n-m+1},\ldots,X_n$ y en forma de straight line program de longitud L en las variables $X_1,\ldots,X_{n-m}$, evaluados en el punto $(\xi_1,\ldots,\xi_{n-m})$ fijado al principio de la demostración.

Cuando aplicamos la primera parte del algoritmo exhibido en el

Teorema 3.2.1. a W obtenemos como salida

$$\left\{ B_j^{(r)} \; ; \; \alpha_j^{\gamma^{(r)}} \; ; \; r_1^{\gamma^{(r)}}, \ldots, r_m^{\gamma^{(r)}} \right\}_{\substack{1\le j\le b \\ 1\le r\le c}}$$

Para cada j y r calculamos como antes los polinomios $P_i^{j,r} \in \mathcal{R}[Z]$ $(1\le i\le s+m)$ que satisfacen las propiedades o y oo del Teorema 3.2.1.

Sea $t \le d^{O(m)}$ una cota para $\deg_Z(P_i^{j,r})$ $(1\le i\le s+m)$.

Sea $G^{j,r} \in \mathcal{R}[Z]$ el polinomio definido por

$$G^{j,r} = \left(\prod_{i=1}^{s'} \left(\alpha_j^{\gamma^{(r)}}\right)^{\delta} \cdot g_i\left(\frac{r_1^{\gamma^{(r)}}}{\alpha_j^{\gamma^{(r)}}} , \ldots , \frac{r_m^{\gamma^{(r)}}}{\alpha_j^{\gamma^{(r)}}} \right) \right)^t$$

Sean j_0 y r_0 como en el Teorema 3.2.1.

Afirmación: Cuando $j = j_0$ y $r = r_0$ y además $W \cap \mathcal{U}' = \emptyset$ o sólo contiene puntos aislados de W entonces el polinomio $G^{j,r}$ satisface:

$$W \cap \mathcal{U}' = \emptyset \text{ si y sólo si } \mathrm{mcd}_Z(P_1^{j,r}, \ldots, P_{s+m}^{j,r}) \mid G^{j,r}$$

En efecto, si $\mathrm{mcd}_Z(P_1^{j,r}, \ldots, P_{s+m}^{j,r}) \nmid G^{j,r}$ entonces $\exists\ \xi \in \bar{\mathcal{R}}$ tal que $\mathrm{mcd}_Z(P_1^{j,r}, \ldots, P_{s+m}^{j,r})(\xi) = 0$ y $G^{j,r}(\xi) \neq 0$ ya que la multiplicidad de ξ como raíz de $G^{j,r}$ es mayor o igual que el grado respecto de la variable Z de $\mathrm{mcd}_Z(P_1^{j,r}, \ldots, P_{s+m}^{j,r})$, para todo $\xi \in \bar{\mathcal{R}}$ que sea raíz de $\mathrm{mcd}_Z(P_1^{j,r}, \ldots, P_{s+m}^{j,r})$.

Luego, $P_i^{j,r}(\xi) = 0 \quad \forall\ 1\le i\le s+m$ y $G^{j,r}(\xi) \neq 0$, de donde resulta que

$$\left(\frac{r_1^{\gamma^{(r)}}(\xi)}{\alpha_j^{\gamma^{(r)}}} , \ldots , \frac{r_m^{\gamma^{(r)}}(\xi)}{\alpha_j^{\gamma^{(r)}}} \right) \in W \cap \mathcal{U}'.$$

Recíprocamente, si $W \cap \mathcal{U}' \neq \emptyset$ sea $\omega \in W \cap \mathcal{U}'$. Por hipótesis ω

es un punto aislado de W y por lo tanto satisface

$$\omega = \left(\frac{r_1^{\gamma^{(r)}}(y(\omega))}{\alpha_j^{\gamma^{(r)}}}, \ \ldots \ , \frac{r_m^{\gamma^{(r)}}(y(\omega)}{\alpha_j^{\gamma^{(r)}}} \right)$$

(donde y es la forma lineal del teorema del elemento primitivo como en el Teorema 3.2.1.). Luego,

$$P_i^{j,r}(y(\omega)) = 0 \quad \forall\ 1 \le i \le s+m \quad y \quad G^{j,r}(y(\omega)) \neq 0$$

En consecuencia, $y(\omega)$ es una raíz de $\mathrm{mcd}_Z(P_1^{j,r},\ldots,P_{s+m}^{j,r}))$ que no es raíz de $G^{j,r}$ de donde $\mathrm{mcd}_Z(P_1^{j,r},\ldots,P_{s+m}^{j,r}) \nmid G^{j,r}$ con lo que concluye la demostración de la afirmación.

Como no podemos determinar j_0 y r_0, continuamos con el algoritmo para todo j,r.

Ahora bien, fijados j y r, $\mathrm{mcd}_Z(P_1^{j,r},\ldots,P_{s+m}^{j,r}) \mid G^{j,r}$ si y sólo si $\exists\ Q_1^{j,r},\ldots,Q_{s+m}^{j,r} \in \mathcal{R}'[Z]$ tales que

$$G^{j,r} = \sum_{i=1}^{s+m} Q_i^{j,r}\ P_i^{j,r} \qquad (6)$$

y $\deg_Z Q_i^{j,r} \le d^{O(m)} + \deg_Z G^{j,r} \le \delta\ s'\ d^{O(m)}$

si y sólo si el sistema lineal no homogéneo con coeficientes en $\mathcal{R}$ determinado por (6) tiene una solución en $\mathcal{R}'$, lo que, usando los mismos argumentos que antes, se traduce en una condición polinomial.

En este caso el sistema tiene a lo sumo $\delta\ s'\ d^{O(m)}$ ecuaciones y $\sum_{i=1}^{s+m}(1+\deg_Z\ Q_i^{j,r}) \le \delta\ s\ s'\ d^{O(m)}$ incógnitas.

Como los coeficientes de la matriz del sistema y de la ampliada son los coeficientes de los polinomios $P_1^{j,r}, \ldots, P_{s+m}^{j,r}$ y $G^{j,r}$, para poder armar el sistema escribimos a $P_1^{j,r}, \ldots, P_{s+m}^{j,r}$ en forma densa en la variable Z tal como se hizo en el Teorema 3.2.1. y luego calculamos los coeficientes de $G^{j,r}$ en dicha variable. Para ello, primero daremos un straight line program para $G^{j,r}$ eliminando las divisiones por $\alpha_j^{\gamma^{(r)}}$ de la siguiente manera:

Sea Y una nueva indeterminada y, para cada i tal que $1 \le i \le s'$ sea

$$\tilde{g}_i = g_i\left(Y.r_1^{\gamma^{(r)}}, \ldots, Y.r_m^{\gamma^{(r)}} \right) \in k[\xi_1, \ldots, \xi_{n-m}, T_{uv}, T_u, Z, Y]_{1 \le u, v \le m}$$

Escribimos a cada $\tilde{g}_i$ en forma densa en la variable Y y, si a_{ik} es el coeficiente de Y^k en $\tilde{g}_i$ entonces

$$G^{j,r} = \left[\prod_{i=1}^{s'} \left(\sum_{k=0}^{\delta} a_{ik} \left(\alpha_j^{\gamma^{(r)}} \right)^{\delta - k} \right) \right]^t$$

De esta manera obtenemos un straight line program para $G^{j,r}$ de longitud $L\ s'\ (s\ \delta)^{O(1)}\ d^{O(m)}$.

Ahora, escribimos a $G^{j,r}$ en forma densa en la variable Z interpolando, lo que finalmente cuesta $L\ (\delta\ s'\ s)^{O(1)}\ d^{O(m)}$.

Por lo tanto, los coeficientes de la matriz del sistema son polinomios en $k[X_1, \ldots, X_{n-m}, T_{uv}, T_u]_{1 \le u, v \le m}$ cuyos grados están acotados por $\delta\ \delta'\ s'\ d'\ d^{O(m)}$, dados por un straight line program de longitud $L\ (s\ s'\ \delta)^{O(1)}\ d^{O(m)}$, evaluados en $(\xi_1, \ldots, \xi_{n-m})$.

Luego, para cada j y r, se tiene una condición polinomial $D_j^{(r)}$ equivalente a $mcd_Z(P_1^{j,r}, \ldots, P_{s+m}^{j,r}) \mid G^{j,r}$ donde aparecen a lo sumo $\delta s' d^{O(m)}$ polinomios de grado acotado por $(\delta s')^{O(1)}\ \delta' d' d^{O(m)}$ dados

por un straight line program de longitud $L\ (s\ s'\ \delta)^{O(1)}\ d^{O(m)}$.

En resumen, si W es del tipo (5) y $W \cap \mathcal{U}'$ tiene a lo sumo puntos aislados de W, combinando las salidas del algoritmo del Teorema 3.2.1 con las condiciones polinomiales recién descriptas, resulta que

$$W \cap \mathcal{U}' = \emptyset \iff \bigvee_{\substack{1 \le j \le b \\ 1 \le r \le c}} \left(B_j^{(r)} \wedge D_j^{(r)} \right)$$

Observemos que en las condiciones $B_j^{(r)}$ y $D_j^{(r)}$ aparecen polinomios en $k[\xi_1,\ldots,\xi_{n-m},T_{ik},T_i,Z_1,Z_2]_{1\le i,k\le m}$ y queremos tener condiciones equivalentes a ellas pero que sólo involucren elementos de $k[\xi_1,\ldots,\xi_{n-m}]$. Como los polinomios que aparecen en dichas condiciones son de grado acotado por $s'\ \delta\ d^{O(m)}$ en las $m^2 + m + 2$ variables T_{ik}, T_i, Z_1, Z_2 $(1\le i,k\le m)$ y están dados por un straight line program de longitud $L\ (s\ s'\ \delta)^{O(1)}\ d^{O(m)}$, utilizando el resultado de [He-Sch, 1982] tal como lo hicimos en el Teorema 3.2.1. obtenemos las condiciones deseadas.

De esta manera, cuando aplicamos a una variedad W del tipo (5) el algoritmo de complejidad secuencial $L^2\ (s\ s'\ \delta)^{O(1)} d^{O(m)}$ que hemos construído, obtenemos como salida una fórmula ψ_W libre de cuantificadores, que es una combinación booleana de fórmulas atómicas del tipo:

$$h_1(\xi_1,\ldots,\xi_{n-m}) = 0 \wedge \ldots\ldots \wedge h_{k'}(\xi_1,\ldots,\xi_{n-m}) = 0 \wedge$$
$$\wedge\ h_{k'+1}(\xi_1,\ldots,\xi_{n-m}) \neq 0 \wedge \ldots\ldots \wedge h_k(\xi_1,\ldots,\xi_{n-m}) \neq 0$$

tal que

a) $|\psi_W| \le L^2\ (s\ s'\ \delta)^{O(1)}\ d^{O(m)}$

b) cada h_i es un polinomio en $k[X_1, \ldots, X_{n-m}]$ de grado acotado por $\delta' d' (s' \delta)^{O(1)} d^{O(m)}$

c) los polinomios h_i vienen dados por medio de un straight line program de longitud $L^2 (s\ s'\ \delta)^{O(1)} d^{O(m)}$.

Además, si $W \cap \mathcal{U}'$ contiene a lo sumo puntos aislados de W, entonces:

$$W \cap \mathcal{U}' = \emptyset \text{ si y sólo si } \psi_W$$

Para cada una de las variedades W_r $(0 \leq r \leq m)$ que definimos en (4) aplicamos el algoritmo anterior obteniendo como salida las fórmulas $\psi_{W_0}, \ldots, \psi_{W_m}$.

Afirmación: $V \cap \mathcal{U} = \emptyset$ si y sólo si $\bigwedge_{r=0}^{m} \psi_{W_r}$

En efecto, como $V \cap \mathcal{U} = \emptyset$ si y sólo si $W_r \cap \mathcal{U}' = \emptyset$ para todo $0 \leq r \leq m$, entonces:

Si $V \cap \mathcal{U} = \emptyset$ entonces $W_r \cap \mathcal{U}' = \emptyset$ $\forall\ 0 \leq r \leq m$ y por lo tanto, en particuar, $W_r \cap \mathcal{U}'$ contiene a lo sumo puntos aislados de W de donde ψ_{W_r} $\forall\ 0 \leq r \leq m$

Recíprocamente, si $\bigwedge_{r=0}^{m} \psi_{W_r}$ entonces ψ_{W_m}

Como W_m por construcción tiene a lo sumo puntos aislados entonces $W_m \cap \mathcal{U}'$ contiene a lo sumo puntos aislados de W_m de donde resulta que $W_m \cap \mathcal{U}' = \emptyset$.

Como λ_m es una forma lineal genérica, $W_{m-1} \cap \{ \lambda_m = 0 \} = W_m$ y $\mathcal{U}'$ es un abierto, por el teorema de la dimensión resulta que

$W_{m-1} \cap \mathcal{U}'$ tiene a lo sumo puntos aislados de W_{m-1} y como satisface $\psi_{W_{m-1}}$, entonces $W_{m-1} \cap \mathcal{U}' = \emptyset$.

Iterando este razonamiento obtenemos que $W_r \cap \mathcal{U}' = \emptyset$ para todo $0 \leq r \leq m$ y por lo tanto $V \cap \mathcal{U} = \emptyset$

Luego, $V \cap \mathcal{U} \neq \emptyset$ si y sólo si $\bigvee_{r=0}^{m} \neg \psi_{W_r} = \psi(\xi_1,\ldots,\xi_{n-m})$

Al igual que en el Teorema 3.2.1., los polinomios que aparecen en la fórmula ψ no dependen del punto $(\xi_1,\ldots,\xi_{n-m})$ fijado,sino que lo que hemos obtenido es una fórmula $\psi(X_1,\ldots,X_{n-m})$ libre de cuantificadores que satisface

$$\{ (x_1,\ldots,x_{n-m}) \in \overline{k}^{n-m} \;/\; \exists\, (x_{n-m+1},\ldots,x_n) \in \overline{k}^{m} :$$
$$F_1(x_1,\ldots,x_n) = 0 \wedge \ldots. \wedge F_s(x_1,\ldots,x_n) = 0 \wedge$$
$$\wedge\, G_1(x_1,\ldots,x_n) \neq 0 \wedge \ldots. \wedge G_{s'}(x_1,\ldots,x_n) \neq 0 \} =$$
$$= \{ (x_1,\ldots,x_{n-m}) \in \overline{k}^{n-m} \;/\; \psi(x_1,\ldots,x_{n-m}) \} \qquad \blacksquare$$

Observación 3.4.2. Sean $d,r \in \mathbb{N}$ tales que $d \geq r \geq 3$ y sea φ la fórmula:

$$\exists\, x_1\, \exists\, x_2 \ldots \exists\, x_{r-1} : x_1^d.y_1 - 1 = 0 \wedge x_2^d.y_2 - x_1 = 0 \wedge$$
$$\wedge\, x_3^d.y_3 - x_2 = 0 \wedge \ldots. \wedge x_{r-1}^d.y_{r-1} - x_{r-2} = 0 \wedge$$
$$\wedge\, y_r^d - x_{r-1} = 0 \wedge y_2.y_r - x_1 \neq 0$$

Es claro que φ es equivalente a la fórmula sin cuantificadores:

$$y_r^{d^r}.y_{r-1}^{d^{r-2}}.y_{r-2}^{d^{r-3}} \ldots.y_2^d.y_1 - 1 = 0 \wedge y_1.y_2^d.y_r^d - 1 \neq 0$$

Sean $g,f \in k[Y_1,\ldots,Y_r]$ los polinomios $g = = Y_1.Y_2^d.Y_r^d - 1$ y

$f = Y_r^{d^r}.Y_{r-1}^{d^{r-2}}.Y_{r-2}^{d^{r-3}}\ldots Y_2^d.Y_1 - 1.$

Como

$$\{(Y_1,\ldots,Y_r) \in \overline{k}^r / f(Y_1,\ldots,Y_r) = 0\} =$$

$$= \{(Y_1,\ldots,Y_r) \in \overline{k}^r / f(Y_1,\ldots,Y_r) = 0 \wedge g(Y_1,\ldots,Y_r) \neq 0\} \cup$$

$$\cup \{(Y_1,\ldots,Y_r) \in \overline{k}^r / f(Y_1,\ldots,Y_r) = 0 \wedge g(Y_1,\ldots,Y_r) = 0\}$$

y los conjuntos

$$\{(Y_1,\ldots,Y_r) \in \overline{k}^r / f(Y_1,\ldots,Y_r) = 0\}$$

y

$$\{(Y_1,\ldots,Y_r) \in \overline{k}^r / f(Y_1,\ldots,Y_r) = 0 \wedge g(Y_1,\ldots,Y_r) = 0\}$$

son cerrados, entonces

$$\{(Y_1,\ldots,Y_r) \in \overline{k}^r / f(Y_1,\ldots,Y_r) = 0\} =$$

$$= \overline{\{(Y_1,\ldots,Y_r) \in \overline{k}^r / f(Y_1,\ldots,Y_r) = 0 \wedge g(Y_1,\ldots,Y_r) \neq 0\}} \cup$$

$$\cup \{(Y_1,\ldots,Y_r) \in \overline{k}^r / f(Y_1,\ldots,Y_r) = 0 \wedge g(Y_1,\ldots,Y_r) = 0\}$$

Además, como $\{(Y_1,\ldots,Y_r) \in \overline{k}^r / f(Y_1,\ldots,Y_r) = 0\}$ es irreducible ya que f es irreducible y distinto de

$$\{(Y_1,\ldots,Y_r) \in \overline{k}^r / f(Y_1,\ldots,Y_r) = 0 \wedge g(Y_1,\ldots,Y_r) = 0\}$$

ya que f no divide a g, entonces debe ser

$$\{(Y_1,\ldots,Y_r) \in \overline{k}^r / f(Y_1,\ldots,Y_r) = 0\} =$$

$$= \overline{\{(Y_1,\ldots,Y_r) \in \overline{k}^r / f(Y_1,\ldots,Y_r) = 0 \wedge g(Y_1,\ldots,Y_r) \neq 0\}}$$

de donde se deduce, aplicando un razonamiento similar al usado en

3.3., que cualquier algoritmo en el modelo de la representación densa de polinomios tendrá una complejidad secuencial no inferior a d^{r^2}.

Sin embargo, aplicando el algoritmo del Teorema 3.4.1. se obtiene una fórmula sin cuantificadores equivalente a φ en tiempo secuencial del orden de $d^{O(r)}$ lo que muestra que, también en el caso en que haya desigualdades, las cotas obtenidas en 3.4.1. mejoran la complejidad secuencial de cualquier algoritmo posible en el modelo de la representación densa de polinomios.

Observación 3.4.3. Sean $X_1,\ldots,X_n$ indeterminadas sobre k, sean $F_1,\ldots,F_s \in k[X_1,\ldots,X_n]$ polinomios cuyos grados totales están acotados por un número natural $d \geq n$ y sean $G_1,\ldots,G_{s'} \in k[X_1,\ldots,X_n]$ polinomios cuyos grados totales están acotados por un número natural δ.

Supongamos que los polinomios $F_1,\ldots,F_s,G_1,\ldots,G_{s'}$ están dados por un straight line program de longitud L.

Sea $\mathcal{P}$ el conjunto definido por

$$\mathcal{P} = \{ \omega \in \overline{k}^n / \; F_1(\omega) = 0 \wedge \ldots \wedge F_s(\omega) = 0 \wedge$$
$$\wedge\, G_1(\omega) \neq 0 \wedge \ldots \wedge G_{s'}(\omega) \neq 0 \}$$

Observemos que la condición "$\mathcal{P}$ es no vacío" puede describirse mediante la fórmula de $\mathcal{L}(k)$

$$\exists\, x_1 \ldots \exists\, x_n : F_1(x_1,\ldots,x_n) = 0 \wedge \ldots \wedge F_s(x_1,\ldots,x_n) = 0 \wedge$$
$$\wedge\, G_1(x_1,\ldots,x_n) \neq 0 \wedge \ldots \wedge G_{s'}(x_1,\ldots,x_n) \neq 0 \}$$

Luego, el utilizando el algoritmo del Teorema 3.4.1., de complejidad secuencuencial $L^2\ (s.s'.\delta)^{O(1)}\ d^{O(n)}$, puede determinarse si el conjunto $\mathcal{P}$ es no vacío.

4. El caso general

En esta sección se exhibirá un algoritmo que resuelve el problema de la eliminación de cuantificadores sobre un cuerpo algebraicamente cerrado para cualquier fórmula. Dado que toda fórmula arbitraria φ puede ser transformada por medio de un algoritmo bien paralelizable de complejidad secuencial $O(|\varphi|)$ en una fórmula prenexa equivalente sin modificar ni $|\varphi|$ ni el grado de los polinomios que aparecen ni la cantidad de variables (ver [Kr, 1990]), asumiremos sin pérdida de generalidad que la fórmula de entrada es prenexa. En 4.1. se construirá un algoritmo que a partir de cualquier fórmula libre de cuantificadores encuentra una equivalente escrita como disyunción de conjunciones de igualdades y desigualdades polinomiales. En 4.2. se resolverá el problema para el caso de un solo bloque de cuantificadores y en 4.3. se resolverá recursivamente el caso general.

4.1. Forma disyuntiva "consistente"

Sean $F_1,\ldots,F_s \in k[X_1,\ldots,X_n]$ y sea $I = \{1,2,\ldots,s\}$

Definición 4.1.1. Se dice que $\mathcal{Z} \subseteq \mathbb{A}^n(\overline{k})$ es una $F_1,\ldots,F_s$-*celda* si:

i) $\mathcal{Z} \neq \emptyset$

ii) $\exists\ M \subseteq I$ tal que

$$\mathcal{Z} = \{x \in \overline{k}^n / \ F_i(x) = 0\ \forall\ i \in M \wedge F_j(x) \neq 0\ \forall\ x \in I-M\}$$

Sea φ una fórmula libre de cuantificadores que es una combina-

ción booleana de fórmulas atómicas que involucran a los polinomios $F_1,\ldots,F_s$, sea $D = \text{máx}\{1 + \sum_{i=1}^{s} \deg F_i\ ,\ n\ ,\ s\}$ y sea $|\varphi|$ la longitud de φ, es decir, el número de símbolos necesarios para codificar a φ. Se construirá un algoritmo que, para $F_1,\ldots,F_s$ dados en forma densa, encuentre una fórmula equivalente a φ que sea una disyunción de conjunciones consistentes (es decir, que definen subconjuntos no vacíos de $\overline{k}^n$) de manera tal que cada una de esas conjunciones consistentes corresponda a una $F_1,\ldots,F_s$-celda. En otras palabras, si $C = \{M \subseteq I \ /\ M$ define una $F_1,\ldots,F_s$-celda$\}$, el algoritmo encontrará un subconjunto S de C tal que φ es equivalente a la fórmula:

$$\bigvee_{M\in S} \bigwedge_{i\in M} F_i(x_1,\ldots,x_n) = 0 \wedge \bigwedge_{j\in I-M} F_j(x_1,\ldots,x_n) \neq 0$$

Esto es posible pues las $F_1,\ldots,F_s$-celdas son los átomos del álgebra de Boole de los subconjuntos de $\overline{k}^n$ definibles por fórmulas libres de cuantificadores que involucran a $F_1,\ldots,F_s$.

Manteniendo las notaciones anteriores, se tiene el siguiente:

Teorema 4.1.2. *A menos de una preparación previa, existe un algoritmo bien paralelizable y sin divisiones de complejidad secuencial* $O(|\varphi|) + D^{O(n)}$ *que a partir de la fórmula* φ *encuentra el subconjunto* S.

Demostración: A lo largo de esta demostración, $[a]$ denotará la parte entera de a y log denotará el logaritmo en base 2.

Procederemos primero a encontrar el conjunto C que caracteriza

a las $F_1,\ldots,F_s$-celdas, lo que se hará en $2+[\log(s-1)]$ etapas.

Para facilitar la notación

$$\{F_{i_1} = 0 \wedge \ldots \wedge F_{i_h} = 0 \wedge F_{i_{h+1}} \neq 0 \wedge \ldots \wedge F_{i_k} \neq 0\}$$

denotará el conjunto algebraico

$$\{x \in \overline{k}^n / F_{i_1}(x) = 0 \wedge \ldots \wedge F_{i_h}(x) = 0 \wedge F_{i_{h+1}}(x) \neq 0 \wedge \ldots \wedge F_{i_k}(x) \neq 0\}$$

Sea $\alpha = 2^{1+\log[(s-1)]}$. Para evitar la separación en casos de acuerdo al desarrollo binario de s, definimos $F_j = 0 \quad (s+1 \leq j \leq \alpha)$.

En la etapa cero, se consideran los conjuntos

$$A_i^0 = \{ \{F_i = 0\}, \{F_i \neq 0\} \} \qquad (1 \leq i \leq \alpha)$$

Utilizando el algoritmo del Teorema 3.4.1. se determina para cada $A_i^0 \quad (1 \leq i \leq \alpha)$ cuáles de sus elementos son el conjunto vacío (notar que los polinomios $F_1,\ldots,F_s$ pueden ser codificados por medio de un straight line program de longitud $D^{O(n)}$ ya que son s polinomios en n variables de grado acotado por D y $s \leq D$).

En la etapa 1 se consideran los conjuntos

$$A_i^1 = \{B \cap C \;/\; B \in A_{2i-1}^0 \wedge C \in A_{2i}^0 \wedge B \neq \emptyset \wedge C \neq \emptyset\} \qquad \left(1 \leq i \leq \frac{\alpha}{2}\right)$$

y, como antes, para cada $A_i^1 \quad \left(1 \leq i \leq \frac{\alpha}{2}\right)$ se determina cuáles de sus elementos son el conjunto vacío.

Suponiendo concluída la etapa j-ésima, en la etapa j+1-ésima se consideran los conjuntos

$$A_i^{j+1} = \{B \cap C \;/\; B \in A_{2i-1}^j \wedge C \in A_{2i}^j \wedge B \neq \emptyset \wedge C \neq \emptyset\} \quad \left(1 \leq i \leq \frac{\alpha}{2^{j+1}}\right)$$

y se decide cuáles de los elementos de $A_i^{j+1} \quad \left(1 \leq i \leq \frac{\alpha}{2^{j+1}}\right)$ son el conjunto vacío.

Observar que si $B \in A_1^j$ entonces B está definido por una conjunción de 2^j fórmulas atómicas que involucran a los polinomios F_k, con $(1+2^j.(i+1) \leq k \leq 2^j.i)$.

Al llegar a la etapa $1+[\log (s-1)]$ quedará definido un único conjunto $A_1^{1+[\log(s-1)]}$ y todos sus elementos involucran a todos los polinomios F_k $(1 \leq k \leq \alpha)$.

Si $B \in A_1^{1+[\log(s-1)]}$ entonces será de la forma:

$$B = \{F_{i_1} = 0 \wedge \ldots \wedge F_{i_h} = 0 \wedge F_{i_{h+1}} \neq 0 \wedge \ldots \wedge F_{i_\alpha} \neq 0\}$$

Definimos $M_B = \{i_1,\ldots,i_h\} \cap \{1,\ldots,s\}$ y con esto descartamos a los polinomios $F_{s+1},\ldots,F_\alpha$. Luego,

$$C = \{M_B \ / \ B \in A_1^{1+[\log(s-1)]} \wedge B \neq \emptyset\}$$

Una vez obtenido el conjunto C, se puede encontrar el conjunto S mediante un algoritmo de complejidad secuencial $O(|\varphi|)$ (ver, por ejemplo, [He, 1983] y [Fi-Ga-Mo, 1990])

Observemos que $\forall\ 0 \leq j \leq 1+[\log(s-1)]$ y $\forall\ 1 \leq i \leq \frac{\alpha}{2^j}$ el conjunto A_i^j tiene a lo sumo D^n elementos (ver [He, 1983], Corollary 1). Dado además que los elementos de los conjuntos de la etapa j+1-ésima se construyen a partir de elementos no vacíos de los conjuntos de la etapa j-ésima, resulta que el algoritmo del Teorema 3.4.1. se utiliza a lo sumo $4sD^{2n}$ veces. Como cada vez que se utiliza dicho algoritmo la complejidad secuencial es del orden de $D^{O(n)}$, ya que $F_1,\ldots,F_s$ son $s \leq D$ polinomios de grado total acotado por D y pueden ser codificados por medio de un straight line program de lon-

gitud $D^{O(n)}$, resulta que la complejidad del algoritmo que hemos construído es del orden de $O(|\varphi|) + D^{O(n)}$ ∎

Observación 4.1.3. En el caso en que los polinomios $F_1,\ldots,F_s$ vengan dados por un straight line program de longitud L, se utilizará el algoritmo descripto en la Proposición 3.1.1. para codificarlos en forma densa y se aplicará luego el algoritmo del Teorema 4.1.2. En este caso la complejidad será del orden de $O(|\varphi|)+L.d^{O(n)}+D^{O(n)}$ donde d es una cota para los grados de los polinomios $F_1,\ldots,F_s$.

4.2. Un solo bloque de cuantificadores

Sea φ una fórmula prenexa con un solo bloque de cuantificadores que involucra s polinomios $F_1,\ldots,F_s \in k[X_1,\ldots,X_n]$ codificados en forma densa y sea $D = \text{máx}\ \{1+\sum_{i=1}^{s} \deg F_i,\ n,\ s\}$. Podemos suponer que los cuantificadores son existenciales (ya que $\forall = \neg\,\exists\,\neg$), es decir, que φ es una fórmula del tipo

$$\exists\, x_{n-m+1} \ \ldots\ \exists\, x_n \ :\ \varphi(x_1,\ldots,x_n)$$

Construimos un algoritmo para eliminar cuantificadores combinando los ya descriptos de la siguiente manera:

Utilizando el Teorema 4.1.2., calculamos primero la forma disyuntiva consistente correspondiente a φ.

Luego, usando que los cuantificadores existenciales conmutan con las disjunciones, aplicando el algoritmo del Teorema 3.4.1. a cada uno de los términos de la disjunción (que son a lo sumo D^n) y

teniendo en cuenta que $s \leq D$ polinomios en n variables de grado total acotado por D pueden codificarse por medio de un straight line program de longitud $D^{O(n)}$, se obtiene una fórmula sin cuantificadores equivalente a la dada.

Por lo tanto, se tiene el siguiente

Teorema 4.2.1. *Sea φ una fórmula prenexa con un solo bloque de cuantificadores que involucra s polinomios $F_1,\ldots,F_s \in k[X_1,\ldots,X_n]$ codificados en forma densa, sea* $D = \text{máx}\ \{1+\sum_{i=1}^{s} \deg F_i,\ n,\ s\}$ *y sea $|\varphi|$ la longitud de φ.*

A menos de una preparación previa, existe un algoritmo bien paralelizable y sin divisiones de complejidad secuencial del orden de $O(|\varphi|) + D^{O(n)}$ que encuentra una fórmula ψ equivalente a φ, libre de cuantificadores. La longitud de la fórmula ψ, así como también la cantidad de polinomios que aparecen en ella, es del orden de $D^{O(n)}$. Además, los polinomios de salida tendrán grados acotados por $D^{O(n)}$ y vendrán dados por un straight line program de longitud $D^{O(n)}$. ∎

Este resultado mejora cualquier cota posible en el modelo de la representación densa de los polinomios (ver 3.4.2.).

4.3. Una fórmula arbitraria

Iterando el algoritmo del Teorema 4.2.1. tantas veces como cuantificadores haya y teniendo en cuenta la Observación 4.1.3. se obtiene el siguiente resultado general:

Teorema 4.3.1. *Sea φ una fórmula prenexa con r bloques de cuantificadores que involucra a s polinomios $F_1,\ldots,F_s \in k[X_1,\ldots,X_n]$ codificados en forma densa, sea* $D = \max\{1+\sum_{i=1}^{s}\deg F_i,\ n,\ s\}$ *y sea $|\varphi|$ la longitud de φ.*

A menos de una preparación previa existe un algoritmo bien paralelizable y sin divisiones de complejidad secuencial del orden de $O(|\varphi|) + D^{(O(n))^r}$ *que encuentra una fórmula ψ equivalente a φ, libre de cuantificadores. La longitud de la fórmula ψ, así como también la cantidad de polinomios que aparecen en ella, es del orden de* $D^{O(n)^r}$. *Además, los polinomios de salida tendrán grados acotados por* $D^{O(n)^r}$ *y vendrán dados por un straight line program de longitud* $D^{O(n)^r}$. ∎

Notar que la complejidad de este algoritmo es mejor que las complejidades de los algoritmos de eliminación conocidos que son del tipo $D^{n^{cr}}$ donde $c \geq 2$ es una constante universal.

Una posibilidad para lograr un algoritmo de eliminación de cuantificadores con mejores cotas de complejidad podría surgir al aplicar los resultados obtenidos en [Gi et al, 1995] que involucran cotas que dependen más intrínsecamente de la geometría del problema. Sin embargo, este análisis excede los alcances del presente trabajo.

5. Una aplicación: Cálculo de la Forma de Chow

A continuación se dará, a modo de aplicación de los algoritmos de eliminación de cuantificadores exhibidos en los Teoremas 3.2.1. y 3.4.1., un algoritmo eficiente para el cálculo del polinomio de Chow de una variedad proyectiva irreducible (ver, por ejemplo, [Ca, 1990]).

Sean k un cuerpo, $\overline{k}$ una clausura algebraica de k y $k[X_0,\ldots,X_n]$ el anillo de polinomios en las indeterminadas $X_0,\ldots,X_n$ a coeficientes en k.

Recordemos brevemente la definición de la Forma de Chow de una variedad proyectiva irreducible.

Sea $\mathbb{P}^n$ el espacio proyectivo n-dimensional sobre $\overline{k}$ y sea

$$\mathcal{X} = \{ x \in \mathbb{P}^n \;/\; F_1(x) = 0 \wedge \ldots \wedge F_s(x) = 0 \}$$

una variedad irreducible de $\mathbb{P}^n$, donde $F_1,\ldots,F_s \in k[X_0,\ldots,X_n]$

Sea r la dimensión proyectiva de $\mathcal{X}$.

Un hiperplano de $\mathbb{P}^n$ se identificará con el punto de $\mathbb{P}^n$ formado por los coeficientes de cualquier forma lineal que lo defina.

Se considera el subconjunto Γ de $(\mathbb{P}^n)^{r+1} \times \mathcal{X}$ definido por:

$$\Gamma = \{(y^{(0)},\ldots,y^{(r)},x) \in (\mathbb{P}^n)^{r+1} \times \mathcal{X} \;/\; x \in y^{(i)} \quad (0\le i\le r)\}$$

y sea

$$\varphi(\Gamma) = \{(y^{(0)},\ldots,y^{(r)}) \in (\mathbb{P}^n)^{r+1} : \exists\, x \in \mathcal{X} \;/\; x \in y^{(i)} \ (0\le i\le r)\}$$

la proyección de Γ a $(\mathbb{P}^n)^{r+1}$.

Entonces, $\varphi(\Gamma)$ resulta ser una hipersuperficie en $(\mathbb{P}^n)^{r+1}$ definible por un polinomio irreducible $F_{\mathcal{X}}$. Este polinomio se llama la Forma de Chow de $\mathcal{X}$ y determina unívocamente a la variedad (ver [Sh], Ch I, § 6).

Se tiene la caracterización

$$x \in \mathcal{X} \Leftrightarrow (x \in y^{(i)} \quad \forall i,\ 0 \le i \le r \ \Rightarrow \ F_{\mathcal{X}}(y^{(0)},\ldots,y^{(r)}) = 0)$$

La importancia de la Forma de Chow es que su grado es el grado de la variedad irreducible $\mathcal{X}$.

En lo que sigue se supondrá que k es efectivo (en el caso en que k tenga característica $p > 0$ la extracción de raíces p-ésimas también se supondrá efectiva).

Teorema 5.1.1. *Sean* $F_1,\ldots,F_s \in k[X_0,\ldots,X_n]$ *polinomios de grado acotado por* $d > n$. *Sea* $\mathcal{X} = \{x \in \mathbb{P}^n \ / \ F_1(x) = 0 \wedge \ldots \wedge F_s(x) = 0\}$ *una variedad proyectiva irreducible y sea* r *su dimensión proyectiva. A menos de una preparación previa, existe un algoritmo bien paralelizable de complejidad secuencial* $s^{O(1)}\ d^{O(nr)}$ *que tiene como entrada a los polinomios* $F_1,\ldots,F_s$ *codificados en forma densa y produce como salida al polinomio de Chow de la variedad* $\mathcal{X}$, *codificado por un straight line program de longitud* $s^{O(1)}\ d^{O(n)}$.

Demostración: Para cada i, $0 \le i \le r$, sea $L^{(i)} = Y_0^i X_0 + \ldots + Y_n^i X_n$, donde Y_j^i $(0 \le i \le r\ ,\ 0 \le j \le n)$ son nuevas indeterminadas sobre $k[X_0,\ldots,X_n]$.

Sea $\Gamma \subseteq (\mathbb{P}^n)^{r+1} \times \mathbb{P}^n$ el conjunto de ceros del ideal generado por $F_1,\ldots,F_s,L^{(0)},\ldots,L^{(r)}$ en $\bar{k}[X_0,\ldots,X_n,Y_j^i]_{0\le i\le r,0\le j\le n}$

Sea $\varphi : (\mathbb{P}^n)^{r+1} \times \mathbb{P}^n \longrightarrow (\mathbb{P}^n)^{r+1}$ la proyección canónica. El conjunto $\varphi(\Gamma)$ es cerrado e irreducible de codimensión 1 (ver [Ne, 1977], Lema 4) y por lo tanto de la forma:

$$\varphi(\Gamma) = \{y \in (\mathbb{P}^n)^{r+1} \ / \ F(y) = 0\}$$

para algún polinomio irreducible $F \in k[Y_j^i]_{0\le i\le r, 0\le j\le n}$ que es el polinomio de Chow de la variedad proyectiva irreducible $\mathfrak{X}$.

Sea $W = \{\omega \in \overline{k}^{(n+1)(r+1)} / \ F(\omega) = 0\}$. Claramente W es la variedad afin correspondiente a $\varphi(\Gamma)$ y por lo tanto

$$W = \{ \ (y_0^0,\ldots,y_n^r) \in \overline{k}^{(n+1)(r+1)} / \ \phi(y_0^0,\ldots,y_n^r) \ \}$$

donde ϕ es la fórmula:

$$\exists x_0 \ldots \exists x_n : F_1(x_0,\ldots,x_n) = 0 \wedge \ldots \wedge F_s(x_0,\ldots,x_n) = 0 \wedge$$

$$\wedge L^{(0)}(x_0,\ldots,x_n,y_0^0,\ldots,y_n^0) = 0 \wedge$$

$$\wedge \ldots \wedge L^{(r)}(x_0,\ldots,x_n,y_0^r,\ldots,y_n^r) = 0$$

que tiene un solo bloque de cuantificadores, n+1 variables ligadas y (n+1)(r+1) variables libres, los polinomios que intervienen están dados en forma densa en las variables $X_0,\ldots,X_n$ y sus coeficientes, que son elementos de $k[Y_j^i]_{0\le i\le r, 0\le j\le n}$ pueden ser evaluados por medio de un straight line program de longitud (r+1)(n+1).

Utilizando el algoritmo del Teorema 3.2.1., obtenemos una fórmula ψ sin cuantificadores equivalente a ϕ. Luego,

$$W = \{ \ \omega \in \overline{k}^{(n+1)(r+1)} / \ \psi(\omega) \ \}$$

Sean $H_1,\ldots,H_k$ los polinomios que aparecen en ψ.

Sean $I = \{1,\ldots,k\}$. Para facilitar la notación, si M es un sub-

conjunto de I

$$\left\{ \bigwedge_{i \in M} H_i = 0 \wedge \bigwedge_{j \in I-M} H_j \neq 0 \right\}$$

denotará el conjunto algebraico

$$\{ \omega \in \overline{k}^{(r+1)(n+1)} / \; H_i(\omega) = 0 \; \forall \; i \in M \wedge H_j(\omega) \neq 0 \; \forall \; j \in I-M \}$$

Análogamente, si $G_1, \ldots, G_h \in k[Y^i_j]_{0 \leq i \leq r, 0 \leq j \leq n}$, dado i / $1 \leq i \leq h$

$$\{ G_1 = 0 \wedge \ldots \wedge G_i = 0 \wedge G_{i+1} \neq 0 \wedge \ldots \wedge G_h \neq 0 \}$$

denotará el conjunto algebraico

$$\{\omega \in \overline{k}^{(r+1)(n+1)} / \; G_j(\omega) = 0 \; \forall \; 1 \leq j \leq i \wedge G_j(\omega) \neq 0 \; \forall \; i+1 \leq j \leq h\}$$

Sea C = {M ⊆ I / M define una $H_1, \ldots, H_k$-celda} (ver Definición 4.1.1.) y sea S el subconjunto de C tal que

$$W = \bigcup_{M \in S} \left\{ \bigwedge_{i \in M} H_i = 0 \wedge \bigwedge_{j \in I-M} H_j \neq 0 \right\}$$

Notar que existe un único subconjunto S de C que satisface esta condición pues las $H_1, \ldots, H_k$-celdas son los átomos del álgebra de Boole de los subconjuntos de $\overline{k}^{(r+1)(n+1)}$ definibles por fórmulas libres de cuantificadores que involucran a $H_1, \ldots, H_k$ y que, además, $\left\{ \bigwedge_{i \in M} H_i = 0 \wedge \bigwedge_{j \in I-M} H_j \neq 0 \right\}$ es no vacío ∀ M ∈ S ya que S es un subconjunto de C.

Como W = { F = 0 } entonces W es cerrado y como

$$W = \bigcup_{M \in S} \left\{ \bigwedge_{i \in M} H_i = 0 \wedge \bigwedge_{j \in I-M} H_j \neq 0 \right\}$$

entonces

$$W = \bigcup_{M \in S} \overline{\left\{ \bigwedge_{i \in M} H_i = 0 \wedge \bigwedge_{j \in I-M} H_j \neq 0 \right\}}$$

W es una variedad irreducible ya que W = { F = 0 } y F es irreducible, por lo tanto $\exists\ M_0 \in S$ tal que

$$W = \overline{\left\{ \bigwedge_{i \in M_0} H_i = 0 \wedge \bigwedge_{j \in I - M_0} H_j \neq 0 \right\}}$$

Sea $J = \{ i \in I \ / \ \{ H_i \neq 0 \} \cap W = \emptyset \}$

Afirmación: $M_0 = J$

En efecto, si $u \in M_0$, entonces

$$W = \overline{\left\{ \bigwedge_{i \in M_0} H_i = 0 \wedge \bigwedge_{j \in I - M_0} H_j \neq 0 \right\}} \subseteq \{ H_u = 0 \}$$

de donde $\{ H_u \neq 0 \} \cap W = \emptyset$ y por lo tanto $u \in J$.

Recíprocamente, sea $u \in J$. Entonces $u \in I$ y $W \cap \{ H_u \neq 0 \} = \emptyset$.

Si $u \notin M_0$ entonces $u \in I - M_0$ y por lo tanto

$$\left\{ \bigwedge_{i \in M_0} H_i = 0 \wedge \bigwedge_{j \in I - M_0} H_j \neq 0 \right\} =$$

$$= \left\{ \bigwedge_{i \in M_0} H_i = 0 \wedge \bigwedge_{j \in I - M_0} H_j \neq 0 \right\} \cap \{ H_u \neq 0 \} \subseteq W \cap \{ H_u \neq 0 \} = \emptyset$$

de donde resulta que $\left\{ \bigwedge_{i \in M_0} H_i = 0 \wedge \bigwedge_{j \in I - M_0} H_j \neq 0 \right\} = \emptyset$, lo que es un absurdo ya que $M_0 \in S$.

Hemos demostrado entonces que $J = M_0$ tal como habíamos afirmado. En particular, como $M_0 \in S$, resulta que $J \neq \emptyset$, ya que si $J = \emptyset$ entonces $\emptyset \in S$ pues $J = M_0$. Luego $\{H_1 \neq 0 \wedge \ldots \wedge H_k \neq 0\}$ es no vacío y $\{ H_1 \neq 0 \wedge \ldots \wedge H_k \neq 0 \} \subseteq W = \{ F = 0 \}$, de donde resulta que $H_1 \ldots H_k . F = 0$ y, como $F \neq 0$ entonces $H_i = 0$ para algún

$i \in I$. Pero esto no puede ocurrir ya que $\{ H_1 \neq 0 \wedge \ldots \wedge H_k \neq 0 \}$ era no vacío.

Dado $P \in k[Y_j^i]_{0\leq i\leq r, 0\leq j\leq n}$ con rad(P) denotaremos al *radical* de P, es decir al polinomio libre de cuadrados que se obtiene multiplicando los factores irreducibles no asociados de P (ver [Ne, 1977]) y, dado un conjunto finito no vacío de polinomios

$$\Lambda = \{P_t \in k[Y_j^i]_{0\leq i\leq r, 0\leq j\leq n} \ / \ t \in L\}$$

con mcd $(P_t, t \in L)$ denotaremos al máximo común divisor entre todos los polinomios de Λ.

Sea $G = \text{rad}(\text{mcd}(H_i, i \in J))$ y sea $H = \prod_{j\in I-J} H_j$

Demostraremos ahora que $F = \dfrac{G}{\text{mcd}(G,H)}$

En efecto, como para todo $i \in J$, $\{ F = 0 \} = W \subseteq \{ H_i = 0 \}$ entonces $F|H_i$ para todo $i \in J$, de donde resulta que $F|G$.

Como $G = \dfrac{G}{\text{mcd}(G,H)} \cdot \text{mcd}(G,H)$ y F es irreducible, entonces F divide a $\dfrac{G}{\text{mcd}(G,H)}$ o F divide a mcd(G,H). Pero si $F|\text{mcd}(G,H)$ entonces $F|H$ y como $H = \prod_{j\in I-J} H_j$ y F es irreducible entonces existe $i \in I - J$ tal que $F|H_i$. Luego,

$$W \cap \{ H_i \neq 0 \} = \{ F = 0 \} \cap \{ H_i \neq 0 \} = \varnothing$$

para algún $i \in I - J$, lo que es una contradicción. Por lo tanto F divide a $\dfrac{G}{\text{mcd}(G,H)}$.

Por otra parte, si $f \in k[Y_j^i]_{0\leq i\leq r, 0\leq j\leq n}$ es un polinomio irredu-

cible que divide a $\frac{G}{mcd(G,H)}$ entonces $f|G$ y $f\nmid H$ (pues G es radical) y por lo tanto, $f|H_i$ $\forall$ $i \in J$ y $f\nmid H$.

Como $\{ f = 0 \} = \{f = 0 \wedge H \neq 0\} \cup \{f = 0 \wedge H = 0\}$ y $\{f = 0\}$ y $\{f = 0 \wedge H = 0\}$ son cerrados, entonces

$$\{ f = 0 \} = \overline{\{f = 0 \wedge H \neq 0\}} \cup \{f = 0 \wedge H = 0\}$$

y como $\{ f = 0 \}$ es irreducible (ya que f es irreducible) entonces $\{ f = 0 \} = \overline{\{f = 0 \wedge H \neq 0\}}$ o $\{ f = 0 \} = \{f = 0 \wedge H = 0\}$. Pero si $\{ f = 0 \} = \{f = 0 \wedge H = 0\}$ entonces $\{ f = 0 \} \subseteq \{ H = 0\}$ lo que no puede ocurrir pues $f\nmid H$. Por lo tanto

$$\{f = 0\} = \overline{\{f = 0 \wedge H \neq 0\}}$$

Además, como $f|H_i$ para todo $i \in J$ y $f\nmid H_j$ para todo $j \in I - J$ ya que $f\nmid H$ y $H = \prod_{j \in I-J} H_j$ entonces se tiene que

$$\{f = 0 \wedge H \neq 0\} \subseteq \left\{ \bigwedge_{i \in J} H_i = 0 \wedge \bigwedge_{j \in I-J} H_j \neq 0 \right\} =$$

$$= \left\{ \bigwedge_{i \in M_0} H_i = 0 \wedge \bigwedge_{j \in I-M_0} H_j \neq 0 \right\}$$

pues $J = M_0$. Por lo tanto,

$$\{f = 0\} = \overline{\{f = 0 \wedge H \neq 0\}} \subseteq \overline{\left\{ \bigwedge_{i \in M_0} H_i = 0 \wedge \bigwedge_{j \in I-M_0} H_j \neq 0 \right\}} = W$$

Luego, $\{ f = 0 \} \subseteq W = \{ F = 0 \}$ de donde $f|F$. Como $\frac{G}{mcd(G,H)}$ es radical (pues G es radical) y hemos probado que todo irreducible que lo divide debe necesariamente dividir a F, entonces resulta que $\frac{G}{mcd(G,H)}$ divide a F. Hemos probado entonces que

$$F = \frac{G}{\mathrm{mcd}(G,H)}$$

Ahora concluiremos con la descripción del algoritmo que calcula la Forma de Chow F.

Dado que los polinomios $H_1,\ldots,H_k$ fueron obtenidos aplicando el algoritmo construido en el Teorema 3.2.1. a la fórmula

$$\exists\, x_0 \ldots \exists\, x_n : F_1(x_0,\ldots,x_n) = 0 \wedge \ldots \wedge F_s(x_0,\ldots,x_n) = 0 \wedge$$
$$\wedge\, L^{(0)}(x_0,\ldots,x_n,y_0^0,\ldots,y_n^0) = 0 \wedge$$
$$\wedge \ldots \wedge L^{(r)}(x_0,\ldots,x_n,y_0^r,\ldots,y_n^r) = 0$$

resulta que la complejidad de este primer paso es del orden de $s^{O(1)}\ d^{O(n)}$. Además $k \leq s^{O(1)}\ d^{O(n)}$ y los polinomios $H_1,\ldots,H_k$ son de grado acotado por $d^{O(n)}$ y están dados por un straight line program de longitud $s^{O(1)}\ d^{O(n)}$.

Para cada $1 \leq i \leq k$, decidimos si $\{ H_i \neq 0 \} \cap W = \emptyset$ para determinar el conjunto J aplicando el algoritmo del Teorema 3.4.1. (ver Observación 3.4.3.) a la fórmula:

$$\exists\, x_0 \ldots \exists\, x_n\ \exists\, y_0^0 \ldots \exists\, y_n^r : F_1(x_0,\ldots,x_n) = 0 \wedge \ldots \wedge$$
$$\wedge\, F_s(x_0,\ldots,x_n) = 0 \wedge L^{(0)}(x_0,\ldots,x_n,y_0^0,\ldots,y_n^0) = 0 \wedge$$
$$\wedge \ldots \wedge L^{(r)}(x_0,\ldots,x_n,y_0^r,\ldots,y_n^r) = 0 \wedge H_i(y_0^0,\ldots,y_n^r)$$

La complejidad secuencial de este paso es $s^{O(1)}\ d^{O(nr)}$.

Finalmente, calculamos $G = \mathrm{rad}(\mathrm{mcd}(H_i,\ i \in J))$, $H = \prod_{j \in I-J} H_j$ y $\mathrm{mcd}(G,H)$ utilizando las técnicas de [Kal, 1988] (es decir haciendo transformaciones genéricas para obtener polinomios mónicos en una variable y aplicando un algoritmo de álgebra lineal que calcula el

máximo común divisor para polinomios de una sola variable). La complejidad total del algoritmo construido es, por lo tanto, del orden de $s^{O(1)}\ d^{O(nr)}$. ∎

Referencias:

[Am, 1989] F. Amoroso, *Tests d'appartenance d'après un théorème de Kollár*, Acad. Sci. Paris, Serie I Math 309 (1989), 691-694.

[Bal et al, 1988] J. L. Balcázar, J. Díaz, J. Gabarró, *Structural complexity* I, EATCS Monographs on Theoretical Computer Science 11, Springer, (1988).

[Be-Yg, 1991] C. Berenstein, A. Yger, *Effective Bezout identities in* $\mathbb{Q}[X_1,\ldots,X_n]$, Acta Math. 166 (1991), 69-120.

[Ber, 1984] S. J. Berkowitz, *On computing the determinant in small parallel time using a small number of processors*, Information Processing Letter, 18 (1984), 147-150.

[Bo, 1977] A. Borodin, *On relating time and space to size and depth*, SIAM J. Comput. 6 (1977), 733-744.

[Bo et al, 1982] A. Borodin, J. von zur Gathen, J. Hopcroft, *Fast parallel matrix and gcd computations*, Proc. 23th Annual Symp. FOCS (1982), 65-71.

[Br, 1987] D. Brownawell, *Bounds for the degrees in the Nullstellensatz*, Ann. Math. Second Series, Vol 126 N^0 3 (1987), 577-591.

[Br, 1989] D. Brownawell, *A prime power version of Nullstellensatz*, manuscrito (1989).

[Bro, 1971] W. S. Brown, *On Euclid's algorithm and the computation of polynomial greatest common divisors*, J. ACM, Vol 18, (1971), 478-504.

[Bro-Tr, 1971] W. S. Brown, J. F. Traub, *On Euclid's algorithm and the theory of subresultants*, J. ACM, Vol 18, (1971), 505-514.

[Ca, 1989] L. Caniglia, *Complejidad de algoritmos en geometría computacional*, Tesis, Universidad de Buenos Aires (1989).

[Ca, 1990] L. Caniglia, *How to compute the Chow Form of an unmixed*

polynomial ideal in single exponential time, AAECC, Springer Verlag (1990).

[Ca-Ga-He, 1988] L. Caniglia, A. Galligo, J. Heintz, *Borne simple exponentielle pour les degrés dans le théorème des zéros sur un corps de caractéristique quelconque*, C.R. Acad. Sci. Paris t. 307, Serie I (1988), 255-258.

[Ca-Ga-He, 1989] L. Caniglia, A. Galligo, J. Heintz, *Some new effectivity bounds in computational geometry*, Proc. 6th Int'l Conf. Applied Algebra, Algebraic Algorithms and Error Correcting Codes, Rome 1988, Springer LN Comput. Sci. 357 (1989), 131-151.

[Ca-Gu-Gu, 1991] L. Caniglia, J. Guccione, J. J. Guccione, *Local membership problems for polynomial ideals*, Proc. Intern. Conf. Effective Methods in Algebraic Geometry MEGA 90, Castiglioncello 1990, T. Mora, C. Traverso, eds., Progress in Mathematics 94, Birkhäuser Verlag (1991), 31-45.

[Ch-Gr, 1983] A. L. Chistov, D.Yu. Grigor'ev, *Subexponential time solving systems of algebraic equations I, II*, LOMI Preprints E-9-83, E-10-83, Leningrad (1983).

[Ch-Gr, 1984] A. L. Chistov, D.Yu. Grigor'ev, *Complexity of quantifier elimination in the theory of algebraically closed fields*, Proc. 11th Symp. MFCS 1984, Springer LM Comp. Sci. 176 (1984), 17-31.

[Ch-Kei] C. C. Chang, H. J. Keisler, *Model theory*, Studies in Logic 73, North Holland (1973).

[Co, 1967] G. E. Collins, *Subresultants and reduced polynomial remainder sequences*, J. ACM, Vol 14 (1967), 128-142.

[Fi-Ga-Mo, 1990] N. Fitchas, A. Galligo, J. Morgenstern, *Precise sequential and parallel complexity bounds for quantifier elimination over algebraically closed fields*, Journal of Pure and Applied Algebra 67 (1990), 1-14.

[Fi-Gi-Smi, 1995] N. Fitchas, M. Giusti, F. Smietanski, *Sur le*

complexité du théorème des zéros, Proceedings of the Second International Conference on Approximation and Optimization, La Habana, 1993, J. Gudatt, ed., Peter Lang Verlag, (1995).

[Gat, 1986] J. von zur Gathen, *Parallel arithmetic computations: a survey*, Proc. 13th Symp. MFCS 1986, Springer LN Comput. Sci. 233 (1986), 93-112.

[Gi et al, 1995] M. Giusti, J. Heintz, J. E. Morais, J. Morgenstern, L. M. Pardo, *Straight-line programs in geometric elimination theory*, Manuscrito (1995).

[Gi-He, 1993] M. Giusti, J. Heintz, *La détermination des points isolés et de la dimension d'une variété algébrique peut se faire en temps polynomial*, Computational Algebraic Geometry and Commutative Algebra, Proceedings of the Cortona Conference on Computational Algebraic Geometry and Commutative Algebra, D. Eisenbud, L. Robbiano, eds., Symposia Matematica, Vol XXXIV, Istituto Nazionale di Alta Matematica, Cambridge University Press (1993).

[Gi-He-Sa, 1993] M. Giusti, J. Heintz, J. Sabia, *On the efficiency of effective Nullstellensatze*, Comput. Complexity 3 (1993), 56-95.

[Gr, 1987] D.Yu. Grigor'ev, *The complexity of the decision for the first order theory of algebraically closed fields*, Math. USRR Izvestija, Vol. 29 N^{o} 2 (1987), 459-475.

[He, 1983] J. Heintz, *Definability and fast quantifier elimination over algebraically closed fields*, Theoret. Comput. Sci. 24 (1983), 279-277.

[He, 1989] J. Heintz, *On the computational complexity of polynomials and bilinear mappings, a survey*, Proc. 5th Intern. Conf. Applied Algebra, Algebraic Algorithms and Error Correcting Codes AAECC 5, Menorca 1987 (L. Huguet, A. Poli eds.), Springer LN Comput. Sci. 356 (1989), 269-300.

[He-Sch, 1982] J. Heintz, C. P. Schnorr, *Testing polynomials which*

are easy to compute, Monographie de l'Enseignement Mathematique, Vol. 30, Impr. Kundig, Geneve (1982), 237-254.

[He-Si, 1981] J. Heintz, M. Sieveking, *Absolute primality of polynomials is decidable in random polynomial time in the number of the variables*, 8th International Colloquium on Automata, Languages and Programming ICALP 81, Springer LNCS 115 (1981), 16-28.

[He-Wü, 1975] J. Heintz, R. Wüthrich, *An efficient quantifier elimination algorithm for algebraically closed fields*, SIGSAM Bull. 9 (1975), 11.

[Ie, 1989] D. Ierardi, *Quantifier elimination in the theory of an algebraically-closed field*, Journal ACM (1989), 138-147.

[Kal, 1988] E. Kaltofen, *Greatest common divisors of polynomials given by straight line programms*, Journal ACM 35 N1 (1988), 231-264.

[Ko, 1988] J. Kollar, *Sharp effective Nullstellensatz*, J. AMS 1 (1988), 963-975.

[Kr, 1990] T. Krick, *Complejidad para problemas de geometría elemental*, Tesis, Universidad de Buenos Aires (1990).

[Kr-Par, 1994] T. Krick, L. M. Pardo, *A computational method for diophantine approximation*, Proc. MEGA'94, Progress in Mathematics, Birkhäuser (1994).

[Ma-Tu, 1995] G. Matera, J. M. Turull, *The space complexity of elimination: upper bounds*, manuscrito (1995).

[Mul, 1986] K. Mulmuley, *A fast parallel algorithm to compute the rank of a matrix over an arbitrary field*, Proc. 18th ACM Symp. Theory of Computing (1986), 338-339.

[Ne, 1977] Yu. V. Nesterenko, *Estimates for the orders of zeros of functions of a certain class and applications in the theory of trascendental numbers*, Math. USSR Izvestija Vol 11 (1977) N^{0} 2. Izvestia Akad. Nauk. SSR Set-Mat. Tom 41 (1977). N^{0} 2.

[Ph, 1988] P. Philippon, *Théorème des zéros effectif d'après J. Kollár*, Seminaire I.H.P. (1988).

[Sh] I. R. Shafarevich, *Algebraic geometry*, Springer (1984).

[St, 1972] V. Strassen, *Berechnung und Programm* I, Acta Inform. 1 (1972), 320-334.

[Sto, 1989] H. J. Stoss, *On the representation of rational functions of bounded complexity*, Theoret. Comput. Sci. 64 (1989), 1-13.

[Tar, 1951] A. Tarski, *A decision method for elementary algebra and geometry*, 2nd ed. Univ. of California Press (1951).

[Wü, 1977] R. Wüthrich, *Ein Quantorenelimination verfahren für die Theorie der algebraisch abgeschlossenen Körper*, Ph.D. Tesis, University of Zurich (1977).