



UNIVERSIDAD CENTRAL DE VENEZUELA  
FACULTAD DE CIENCIAS  
ESCUELA DE MATEMÁTICA

# Resolución de algunos problemas de álgebra lineal a través de algoritmos no deterministas

Trabajo Especial de Grado presentado ante la ilustre Universidad Central de Venezuela por el **Br. Mike R. Paolini M.** para optar al título de Licenciado en Matemática.

**Tutor: M.Sc. Edwin Pin.**

**Co-Tutor: Dr. Jean Liendo.**

Caracas, Venezuela

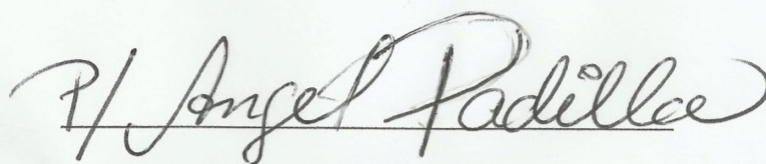
Octubre 2019

## ABSTRACTO

En este trabajo especial de grado se presentan 4 problemas matriciales asociados al rango y al determinante, de donde se muestran clasificaciones de los mismos dependiendo del conjunto a considerar; y la clasificación se realiza en términos del tiempo de resolución a través de algoritmos probabilísticos, o mostrando que no existe algoritmo capaz de resolverlo. Para esto se desarrollan los algoritmos pertinentes al problema, o mediante el uso de propiedades de varios problemas de decisión conocidos y reducciones entre problemas de decisión se muestra una equivalente entre distintos problemas; para algunos de los resultados haremos uso de la teoría existencial de primer orden para sus demostraciones.

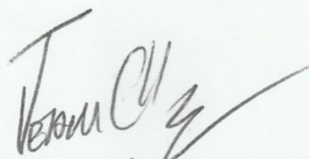
**Palabras Clave:** Matrices, problemas de decisión, algoritmos, clasificación, reducción de problemas.

Nosotros, los abajo firmantes, designados por la Universidad Central de Venezuela como integrantes del Jurado Examinador del Trabajo Especial de Grado titulado “**Resolución de algunos problemas de álgebra lineal a través de algoritmos no deterministas**”, presentado por el **Br. Mike R. Paolini M.**, titular de la Cédula de Identidad **20363615**, certificamos que este trabajo cumple con los requisitos exigidos por nuestra Magna Casa de Estudios para optar al título de **Licenciado en Matemática**.



**Edwin Pin**

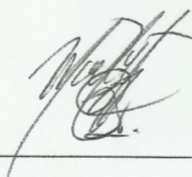
**Tutor**



---

**Jean Carlos Liendo**

**Co-tutor**



---

**Mairene Colina**

**Jurado**



---

**Robinson Rivas**

**Jurado**

## Agradecimiento

Antes de todo quiero agradecer a la persona que más ayuda me ha dado para la realización de este trabajo y a la que le estoy más agradecido, a mi tutor, Edwin Pin.

Quiero agradecer a Jean Carlos Liendo por aceptar ser mi co-tutor.

Gracias a Keisbeth Matamoros por la ayuda y consejos que me dio para poder salir rápido de mis dudas sobre LaTeX.

Gracias a Gabriele Troncone, a Daniela Stefanelli y a Juan Fajardo por el apoyo y ayuda que me dieron.

Gracias a Jedariana Belisario por la guía en la escritura del algoritmo del capítulo 4.

Gracias a mis padres, por el sustento y el apoyo que me han dado toda mi vida.

Quiero dar un agradecimiento a todos los profesores que fueron parte importante de mi formación académica: Ling Sequera, Yeyreimi Freitas, Andrés Pérez, Irene Santos, Manuel Maia, Elio Mendez, Douglas Jiménez, Carmen Da Silva, Juan Sequera, Edwin Pin, Carlos Gonzalez, Mercedes Arrijoja, Tomás Guardia, Juan Guevara, Angie Pineda y Cristina Balderrama.

## Índice general

Introducción	1
<b>Capítulo 1. Preliminares</b>	<b>3</b>
1. Álgebra	3
2. Lógica	4
3. Complejidad Computacional	9
<b>Capítulo 2. Problemas Matriciales</b>	<b>17</b>
1. Introducción	17
2. Algunos ejemplos	19
3. Presentación de los problemas	20
4. Resumen de resultados	21
<b>Capítulo 3. Jerarquía de los Problemas</b>	<b>24</b>
1. Computar MAXRANK sobre cuerpos infinitos	24
2. Problema SING sobre un cuerpo algebraicamente cerrado	26
<b>Capítulo 4. Indecibilidad a través del determinante</b>	<b>30</b>
1. Universalidad del determinante	30
2. Problema SING sobre $\mathbb{Z}$	40
<b>Capítulo 5. Teoría Existencial y Reducciones</b>	<b>42</b>
1. Teoría existencial de primer orden	42
2. Problemas de decisión sobre un cuerpo finito	43
3. Cota inferior para SING sobre $\mathbb{Q}$ y $\mathbb{R}$	46
4. Cota inferior para MINRANK sobre un cuerpo	48
5. Cotas superiores para MINRANK sobre un cuerpo	53
<b>Capítulo 6. Ocurrencia de las variables una única vez</b>	<b>59</b>

Apéndice A. Pseudo-Códigos de las demostraciones	63
Bibliografía	72

## Introducción

Desde la invención de las computadoras, estas se han utilizado para resolver una gran variedad de problemas, que sin ellas no se hubieran podido solucionar o por lo menos no con la misma eficiencia. Por ejemplo, muchos de estos problemas requieren una implementación automatizada de sus soluciones analíticas para poder conseguir una buena aproximación. Uno de tales problemas es el de hallar las raíces de un polinomio de grado alto.

A lo largo de la corta historia de las computadoras (como objetos electrónicos desarrollados en la primera mitad del siglo pasado), tanto matemáticos como computistas se han interesado en crear preguntas y buscar solución a una gran cantidad de problemas.

El objetivo es resolver estos problemas y luego categorizarlos en base al tiempo, almacenamiento o algún otro recurso utilizado. Gracias a esto se puede saber qué problemas pueden resolverse más eficientemente, y en base a eso establecer criterios de complejidad para los mismos. De esta forma nos muestran en [1], [2] y [5] un listado de problemas; algunos algoritmos que los resuelven, su ubicación en la jerarquía polinomial y las posibles relaciones que existen entre ellos. Algunos de estos problemas son de interés puramente matemático, como poder resolver el décimo problema de Hilbert [1], o hallar emparejamientos perfectos para un conjunto de nodos [2, 5]. Existen otros problemas cuya solución da más aporte para solventar o mejorar dificultades de la vida diaria, como determinar la vía más eficiente para el recorrido de un transporte.

En este trabajo de grado se introducen los conceptos básicos de la Teoría de Complejidad Computacional como Máquina de Turing (determinista y no determinista), problemas de decisión, clases de complejidad y reducción entre problemas de decisión; y algunos conceptos de álgebra matricial. En esta última, nos enfocaremos en los conceptos de rango (máximo y mínimo) y singularidad de una matriz. Se mostrará la dificultad que posee calcular el rango de una matriz con coeficientes variables y cómo esta puede verse alterada en base al conjunto que se esté utilizando para las entradas de la misma.

La meta de este trabajo es analizar los cuatro problemas de decisión principales del artículo *The Computational Complexity Of Some Linear Algebra Problems* [1], mostrar la ubicación de estos en la jerarquía polinomial y en clases de complejidad probabilísticas, usando principalmente algoritmos de Monte Carlo.

Como objetivo secundario se buscará usar reducciones para heredar propiedades obtenidas de los cuatro problemas en [1] y se usarán reducciones que involucran resultados ya conocidos de complejidad computacional como el Teorema de Cook, que nos dice que el problema de decisión **SAT** es **NP**-fuerte [6]. Además se analizarán y explicarán las demostraciones de las respectivas reducciones. Para lograr esto haremos uso de la lógica proposicional, la lógica existencial de primer orden y la teoría de grafos.



## Capítulo 1

### Preliminares

En este capítulo presentaremos la base teórica requerida para la lectura de este trabajo especial de grado. Mostraremos definiciones y aspectos básicos de la teoría de la complejidad computacional y de la lógica matemática, en particular, la lógica existencial de primer orden. También se mostrarán algunas definiciones de Álgebra pertinentes para el desarrollo de algunas partes de este trabajo.

#### 1. Álgebra

Haremos uso de polinomios de una y varias variables a lo largo de este trabajo. Por ejemplo  $P(x) = x^4 + 3x - 1$  o  $Q(x, y, z) = 2xy^2 - z^5 + 7x$ . Parte de esto debido a que trabajaremos con determinantes de matrices los cuales son representados con polinomios como en los ejemplos.

**Lema 1.1:** ([4], **Página 211**) Un polinomio (de una variable) de grado  $n$  sobre un cuerpo tiene cuando más  $n$  raíces en cualquier cuerpo extensión.

**Definición 1.2:** Una *Ecuación diofántica* es una ecuación de dos o más variables cuyos coeficientes pertenecen al conjunto de los enteros; y su solución se estudia en los enteros.

Uno de los problemas centrales en el estudio de las ecuaciones diofánticas es el siguiente:

**Décimo problema de Hilbert:** Idear un algoritmo que resuelva una ecuación diofántica cualquiera en un número finito de pasos.

En 1970 se logró resolver el Décimo problema de Hilbert, llegando a la conclusión que no existe tal algoritmo. Por lo cual el problema de hallar una solución a una ecuación diofántica cualquiera es indecidible. (Más adelante en este capítulo se explicará que significa que un problema sea indecidible)

**Definición 1.3:** Sea un polinomio multivariado  $p(x_1, \dots, x_t)$  escrito en su forma desarrollada, es decir, como una suma de monomios. Su *grado total* es el máximo valor de la suma de los exponentes de las variables de sus monomios.

**Ejemplo:**

Sea  $P(x, y, z) = 4xy^2z^3 - 5x + yz^2$ , su grado total es 6.

**Definición 1.4:** Un polinomio  $p(x_1, \dots, x_t)$  se dice que es *multi-afín* sobre un cuerpo  $\mathbb{F}$ , si cualquier substitución de las variables excepto una (digamos  $x_i$ ), por elementos del cuerpo; puede expresarse en la forma  $a \cdot x_i + b$ ;  $a, b \in \mathbb{F}$ .

Un resultado sobre este tipo de polinomios es: un polinomio es multi-afín, si y solo si, todas las variables que aparecen en él lo hacen con grado 1 a lo sumo.

El polinomio en el ejemplo anterior no es multi-afín. En el siguiente ejemplo mostraremos un polinomio que si es multi-afín.

**Ejemplo:**

$$p(x, y, z) = 2xy - 3z + 1.$$

**Definición 1.5:** Un cuerpo  $\mathbb{F}$  se dice *algebraicamente cerrado* si cada polinomio cuyo grado sea 1 o mayor, con coeficientes en  $\mathbb{F}$ , tiene al menos un cero en  $\mathbb{F}$ .

A lo largo de este trabajo usaremos continuamente el siguiente cuerpo:

**Definición 1.6:** Un *cuerpo de Galois* (o cuerpo finito) se define como un cuerpo con una cantidad finita de elementos.

El *orden* de un cuerpo de Galois es el número de elementos que posee. Se cumple la propiedad: Un cuerpo de Galois tiene orden  $q$ , si y solo si,  $q$  se puede escribir como  $p^k$ ; donde  $p$  es un número primo y  $k$  un número entero positivo. Denotaremos los cuerpos de Galois como  $G(q)$ .

**Observación:** Solo hay un cuerpo de Galois de orden  $q$ , salvo isomorfismos.

Los resultados mostrados en el trabajo están relacionados con  $G(q)$ ,  $\mathbb{Q}$ ,  $\mathbb{R}$ ,  $\mathbb{C}$  y  $\mathbb{Z}$ . Los cuerpos de Galois son ampliamente usados en distintas áreas de las matemáticas y ciencias de la computación, como en teoría de números, geometría algebraica, criptografía, entre otras.

## 2. Lógica

En esta sección mostraremos las definiciones necesarias para poder construir y entender fórmulas existenciales de primer orden. Empezaremos explicando la lógica Booleana y su respectiva notación, luego introduciremos la lógica de predicados o lógica de primer orden.

Los símbolos que usaremos en la lógica Booleana son:

Constantes	$0, 1$
Variables	$x_1, x_2, \dots$
Conectivos lógicos	$\vee, \wedge, \neg, \longrightarrow, \longleftrightarrow$
Paréntesis	$(, )$

**Definición 1.7:** Una *variable Booleana* es aquella que solo puede tomar uno de dos valores posibles: 0 o 1. Entretanto, identificaremos a estas variables por las letras  $x, y$  o  $z$  y en caso de ser necesario utilizaremos sub-índices:  $x_1, x_2, x_3, \dots, x_n$ .

Las *fórmulas Booleanas* se definen inductivamente:

- Toda variable Booleana es una fórmula Booleana;
- Si  $\phi$  es una fórmula Booleana, entonces  $\neg\phi$  es una fórmula Booleana a la cual llamamos *negación* de  $\phi$ ;
- Si  $\phi_1$  y  $\phi_2$  son fórmulas Booleanas entonces  $\phi_1 \wedge \phi_2$  es una fórmula Booleana a la cual llamamos *conjunción* de  $\phi_1$  y  $\phi_2$ ;
- Si  $\phi_1$  y  $\phi_2$  son fórmulas Booleanas entonces  $\phi_1 \vee \phi_2$  es una fórmula Booleana a la cual llamamos *disyunción* de  $\phi_1$  y  $\phi_2$ .

En la definición anterior no se muestran los condicionales  $\longrightarrow, \longleftrightarrow$  debido a que podemos escribirlos en términos de los conectivos  $\wedge, \vee$ . Tenemos que la expresión  $x \longrightarrow y$  la podemos escribir como  $\neg x \vee y$  y la expresión  $x \longleftrightarrow y$  la podemos escribir como  $(\neg x \vee y) \wedge (\neg y \vee x)$ .

**Definición 1.8:** Sea  $\phi(x_1, \dots, x_n)$  una fórmula Booleana y  $S \in \{0, 1\}^n$  una asignación de valores de las variables  $x_1, \dots, x_n$ . Decimos que  $S$  *satisface*  $\phi$  si:

- $\phi = x$ , es decir, una variable. Entonces si  $S = 1$  se satisface  $\phi$ .
- $\phi = \neg x$ . Entonces si  $S = 0$  se satisface  $\phi$ .
- $\phi = \phi_1 \vee \phi_2$ . Entonces si  $S$  satisface  $\phi_1$  o  $\phi_2$  entonces se satisface  $\phi$ .
- $\phi = \phi_1 \wedge \phi_2$ . Entonces si  $S$  satisface  $\phi_1$  y  $\phi_2$  entonces se satisface  $\phi$ .

**Ejemplo 1.9:** Si tenemos la fórmula  $\phi \equiv (x \vee y) \longrightarrow z$ , donde  $x, y, z$  pueden tomar el valor 0 (falso) o el valor 1 (verdadero), una posible asignación de valor es  $(x, y, z) = (1, 1, 1)$ . Esta asignación hace que  $\phi$  sea verdadera.

**Definición 1.10:** Sean  $\phi_1$  y  $\phi_2$  un par de fórmulas Booleanas. Decimos que  $\phi_1$  y  $\phi_2$  son *equivalentes*, y lo denotamos por  $\phi_1 \equiv \phi_2$ , si para toda asignación de valores  $S$  (común a ambas fórmulas) se cumple que  $S$  satisface  $\phi_1$ , si y solo si,  $S$  satisface  $\phi_2$ .

**Definición 1.11:** Un *literal positivo* es una variable Booleana y un *literal negativo* es la negación de una variable Booleana. Cuando no sea relevante el signo, diremos simplemente *literal*. Una *cláusula* es una disyunción de literales y un *implicante* es una conjunción de literales.

Una fórmula Booleana que sea una conjunción de cláusulas se dice que está en *forma normal conjuntiva* (FNC) y una fórmula Booleana que sea una disyunción de implicantes se dice que está en *forma normal disyuntiva* (FND).

**Proposición 1.12:** Toda fórmula Booleana es equivalente a una en FNC y a una en FND.

Por la Definición 1.8, podemos comprender cuando una fórmula Booleana es *satisfacible*. Decimos que la fórmula Booleana  $\phi(x_1, \dots, x_n)$  es satisfacible, si existe una asignación  $S \in \{0, 1\}^n$  tal que  $S$  satisface  $\phi$ , esto lo representaremos de la forma  $S \models \phi$ .

A partir de este punto estaremos explicando ya no la lógica usando variables Booleanas sino que usaremos predicados.

Los símbolos que usaremos son:

Constantes	$c_1, c_2, \dots$
Variables	$x_1, x_2, \dots$
Símbolo de igualdad	$=$
Símbolos de funciones	$f_1^{v_1}, f_2^{v_2}, \dots$
Relaciones	$R$
Conectivos lógicos	$\vee, \wedge, \neg, \longrightarrow, \longleftrightarrow$
Cuantificadores	$\exists, \forall$
Paréntesis	$(, )$

**Definición 1.13:** Un *vocabulario* es una tupla de símbolos  $\langle R_1^{a_1}, \dots, R_r^{a_r}, f_1^{v_1}, \dots, f_q^{v_q}, c_1, \dots, c_s \rangle$ , donde cada  $a_i$  y  $v_j$  son un número entero positivo fijo, cada  $R_i$  se dice que es un *símbolo de relación de aridad  $a_i$* , cada  $f_j$  se dice que es un *símbolo de función de aridad  $v_j$*  y cada  $c_k$  se dice que es un *símbolo de constante*. Sea  $\text{Var} = \{x_1, x_2, x_3, \dots\}$  un conjunto numerable de variables, sobre un dominio por definir (estas variables no son variables Booleanas). Un

*término* es o un elemento de  $\text{Var}$  o un símbolo de constante o una expresión de la forma  $f(t_1, \dots, t_v)$ , donde  $f$  es un símbolo de función de aridad  $v$  y  $t_1, \dots, t_v$  son términos.

A los símbolos de relación también se les suele llamar *predicados* y aquí también adoptamos esa terminología.

**Definición 1.14:** Sea  $\sigma$  un vocabulario y  $R$  un predicado de  $\sigma$  de aridad  $a$ . Una expresión de la forma  $R(t_1, \dots, t_a)$ , donde  $t_1, \dots, t_a$  son términos, se denomina *fórmula atómica*.

A una fórmula atómica o a la negación de una fórmula atómica también se le llama *literal*, y se dice que es positivo o negativo según el signo.

Principalmente trabajaremos con la lógica de primer orden. En la lógica de primer orden existen los llamados *cuantificadores*, está el *cuantificador existencial*, denotado por  $\exists$  y el *cuantificador universal*, denotado por  $\forall$ . Donde  $\exists x : \phi$  representa que existe una sustitución  $a$  para  $x$  en el conjunto solución tal que  $a$  satisface a  $\phi$ , mientras que  $\forall x : \phi$  representa que para cualquier sustitución  $a$  de  $x$ , se tiene que  $a$  satisface  $\phi$ .

**Definición 1.15:** Las *fórmulas de primer orden* se definen de manera inductiva:

- Toda fórmula atómica es una fórmula de primer orden.
- Si  $\phi$  es una fórmula de primer orden, entonces  $\neg\phi$  también lo es.
- Si  $\phi_1$  y  $\phi_2$  son fórmulas de primer orden, entonces  $\phi_1 \vee \phi_2$  y  $\phi_1 \wedge \phi_2$  también lo son.
- Si  $\phi$  es una fórmula de primer orden, entonces  $\exists x\phi$  y  $\forall x\phi$  también lo son.

Todas las variables que están al alcance de un cuantificador en una fórmula de primer orden se dice que son *variables cuantificadas*, de lo contrario se dice que son *variables libres*.

La otra definición que nos interesa es la de *sub-fórmula*, que definiremos de manera inductiva igual que como hicimos con la definición de fórmula de primer orden. Antes de eso, es bueno mencionar que una sub-fórmula de  $\Phi$  es cualquier fórmula usada para la construcción de  $\Phi$ .

- $\Phi$  es una sub-fórmula  $\Phi$ .
- Si  $\neg\Psi$  es una sub-fórmula de  $\Phi$ , entonces  $\Psi$  también es una sub-fórmula de  $\Phi$ .
- Si  $\Theta \wedge \Psi$  es una sub-fórmula de  $\Phi$ , entonces  $\Theta$  y  $\Psi$  también son sub-fórmulas de  $\Phi$ .
- Si  $\Theta \vee \Psi$  es una sub-fórmula de  $\Phi$ , entonces  $\Theta$  y  $\Psi$  también son sub-fórmulas de  $\Phi$ .
- Si  $(\forall v_i)\Psi$  es una sub-fórmula de  $\Phi$  y  $v_i$  es una variable, entonces  $\Psi$  también es una sub-fórmula de  $\Phi$ .

- Si  $(\exists v_i)\Psi$  es una sub-fórmula de  $\Phi$  y  $v_i$  es una variable, entonces  $\Psi$  también es una sub-fórmula de  $\Phi$ .

En [8] se trabaja a mayor profundidad lo tratado en esta sección. Para este trabajo, nos concentraremos en trabajar con un tipo específico de fórmula de primer orden.

**Definición 1.16:** Una *sentencia* es una fórmula donde todas las variables que aparecen en ella se encuentran cuantificadas.

Una sentencia se encuentra en *forma normal prenexa* si todos los cuantificadores se encuentran en el extremo izquierdo de la sentencia (es decir, al inicio de la misma). En general, trabajaremos con sentencias en forma normal prenexa.

**Definición 1.17:** Sea el vocabulario  $\sigma = \langle R_1^{a_1}, \dots, R_r^{a_r}, f_1^{v_1}, \dots, f_q^{v_q}, c_1, \dots, c_s \rangle$ . Una  $\sigma$ -estructura es una tupla  $\mathcal{A} = \langle |\mathcal{A}|, R_1^{\mathcal{A}}, \dots, R_r^{\mathcal{A}}, f_1^{\mathcal{A}}, \dots, f_q^{\mathcal{A}}, c_1^{\mathcal{A}}, \dots, c_s^{\mathcal{A}} \rangle$ , donde:

- $|\mathcal{A}|$  es un conjunto no vacío denominado *dominio* de  $\mathcal{A}$ ;
- $R_i^{\mathcal{A}} \subseteq |\mathcal{A}|^{a_i}$ , para todo  $i = 1, \dots, r$ ;
- $f_j^{\mathcal{A}} : |\mathcal{A}|^{v_j} \rightarrow |\mathcal{A}|$ , para todo  $j = 1, \dots, q$ ;
- $c_k^{\mathcal{A}} \in |\mathcal{A}|$ , para todo  $k = 1, \dots, s$ .

Cuando el vocabulario no sea relevante diremos simplemente que  $\mathcal{A}$  es una *estructura*.

Dado un vocabulario  $\sigma$ , denotamos por  $\mathcal{L}[\sigma]$  al lenguaje de fórmulas de primer orden sobre  $\sigma$ . Mostraremos ahora qué significa que una fórmula  $\varphi$  de  $\mathcal{L}[\sigma]$  sea cierta, dada una  $\sigma$ -estructura  $\mathcal{A}$ .

**Definición 1.18:** Sea el vocabulario  $\sigma = \langle R_1^{a_1}, \dots, R_r^{a_r}, f_1^{v_1}, \dots, f_q^{v_q}, c_1, \dots, c_s \rangle$ ,  $\mathcal{A}$  una  $\sigma$ -estructura y  $\varphi$  una fórmula de  $\mathcal{L}[\sigma]$ . Consideremos una función  $\iota : \text{Var} \rightarrow |\mathcal{A}|$ . Esta función se puede extender naturalmente al conjunto de términos sobre  $\sigma$ . Por conveniencia, digamos que  $\iota(c_k) = c_k^{\mathcal{A}}$  para todo  $k = 1, \dots, s$ . Y si  $t = f(t_1, \dots, t_v)$ , entonces  $\iota(t) = f^{\mathcal{A}}(\iota(t_1), \dots, \iota(t_v))$ . A  $\iota$  la denominamos *interpretación*. Describiremos qué significa que  $\mathcal{A}$ , junto con la interpretación  $\iota$ , *satisfaga* la fórmula  $\varphi$ , denotándolo por  $(\mathcal{A}, \iota) \models \varphi$ .

- Si  $\varphi$  es la fórmula atómica  $R_i(t_1, \dots, t_{a_i})$ , con  $i = 1, \dots, r$ , entonces  $(\mathcal{A}, \iota) \models \varphi$ , si y solo si,  $(\iota(t_1), \dots, \iota(t_{a_i})) \in R_i^{\mathcal{A}}$ .
- Si  $\varphi$  es de la forma  $\neg\varphi_1$ , entonces  $(\mathcal{A}, \iota) \models \varphi$ , si y solo si, no es el caso que  $(\mathcal{A}, \iota) \models \varphi_1$  (esto último se denota por  $(\mathcal{A}, \iota) \not\models \varphi_1$ ).

- Si  $\varphi$  es de la forma  $(\varphi_1 \wedge \varphi_2)$ , entonces  $(\mathcal{A}, \iota) \models \varphi$ , si y solo si,  $(\mathcal{A}, \iota) \models \varphi_1$  y  $(\mathcal{A}, \iota) \models \varphi_2$ .
- Si  $\varphi$  es de la forma  $(\varphi_1 \vee \varphi_2)$ , entonces  $(\mathcal{A}, \iota) \models \varphi$ , si y solo si,  $(\mathcal{A}, \iota) \models \varphi_1$  o  $(\mathcal{A}, \iota) \models \varphi_2$ .
- Si  $\varphi$  es de la forma  $\exists x\varphi_1$ , entonces  $(\mathcal{A}, \iota) \models \varphi$ , si y solo si,  $(\mathcal{A}, \iota_{x=a}) \models \varphi_1$  para algún  $a \in |\mathcal{A}|$ . La interpretación  $\iota_{x=a}$  es la misma  $\iota$  salvo que la variable  $x$  se reinterpreta con el valor  $a$ .
- Si  $\varphi$  es de la forma  $\forall x\varphi_1$ , entonces  $(\mathcal{A}, \iota) \models \varphi$ , si y solo si,  $(\mathcal{A}, \iota_{x=a}) \models \varphi_1$  para todo  $a \in |\mathcal{A}|$ .

Si  $\varphi$  es una sentencia, no hace falta definir una interpretación  $\iota$ . En este caso escribimos  $\mathcal{A} \models \varphi$ , si  $\varphi$  se satisface en la estructura  $\mathcal{A}$ .

**Definición 1.19:** Sea  $\sigma$  un vocabulario y  $\varphi_1, \varphi_2 \in \mathcal{L}$  un par de sentencias. Decimos que  $\varphi_1$  y  $\varphi_2$  son *equivalentes*, denotado por  $\varphi_1 \equiv \varphi_2$ , si para toda  $\sigma$ -estructura  $\mathcal{A}$  se cumple que  $\mathcal{A} \models \varphi_1$ , si y solo si,  $\mathcal{A} \models \varphi_2$ .

Existe un resultado parecido al mostrado en la Proposición 1.12, pero para la sentencias en forma normal prenexa.

**Proposición 1.20:** Toda sentencia es equivalente a una sentencia en forma normal prenexa.

### 3. Complejidad Computacional

En esta sección mostraremos los aspectos fundamentales de la complejidad computacional, además de adentrarnos en aspectos de complejidad computacional probabilística. Primero, hay que empezar con la estructura más importante, que a pesar de que no se use directamente en este trabajo, se encontrará presente de manera tácita en la mayoría de las partes del trabajo. Esta definición es la de *Máquina de Turing*, la cual presentaremos formalmente, pero antes daremos una idea intuitiva de la misma. Una Máquina de Turing (MT) se puede interpretar como un modelo computacional que se encarga de leer una palabra (entrada) y escribir una respuesta (salida). La Figura 1 ilustra esta idea.

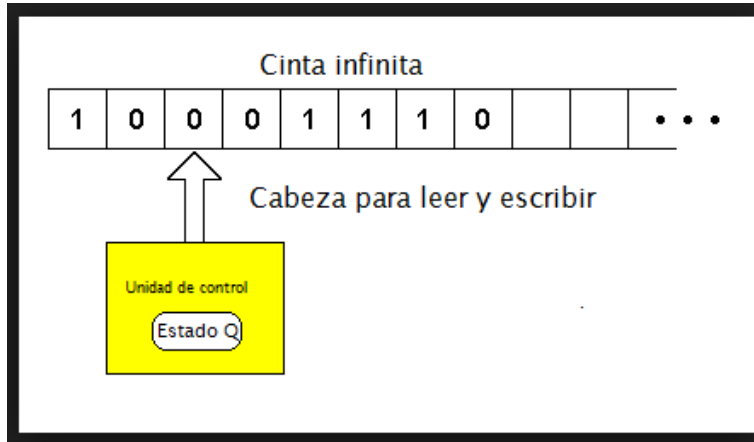


FIGURA 1. Ejemplo de una Máquina de Turing.

**Definición 1.21:** Una *Máquina de Turing* se define como la tupla  $M = (K, \Sigma, \delta, s)$ , donde:

- $K$  = Conjunto finito de estados.
- $\Sigma$  = Conjunto finito de símbolos (alfabeto de  $M$ ).
- $\delta$  = Relación de transición.
- $s$  = Estado inicial.

Se asumirá que  $K$  y  $\Sigma$  son conjuntos disjuntos.  $\Sigma$  siempre contendrá los símbolos especiales  $\sqcup$  y  $\triangleright$ , el primero es el espacio en blanco y el segundo es el primer símbolo con el cual comienzan todas las MT. Por último,  $\delta$  es una relación de transición que está contenido en  $(K \times \Sigma) \times ((K \cup \{h, \text{"yes"}, \text{"no"}\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\})$ .

El conjunto  $\{h, \text{"yes"}, \text{"no"}\}$  representa los estados de parada de una MT. Si tenemos una máquina de Turing  $M$  e introducimos la entrada  $x$  (consideramos una máquina cualquiera y  $x$  una cadena de caracteres usando los símbolos de  $\Sigma$ ). Si la máquina se detiene denotaremos su salida como  $M(x)$ . Si obtenemos  $M(x) = \text{"no"}$  entonces la máquina rechaza la entrada  $x$ , si es  $M(x) = \text{"yes"}$  entonces la máquina acepta la entrada  $x$ . En el caso de tener el estado  $h$ , se obtiene  $M(x) = y$ , donde  $y$  es la cadena de símbolos en  $M$  al momento de la parada (es decir, no se logró llegar a una respuesta de sí o no). Es posible que  $M$  nunca se detenga en la entrada  $x$ , en esos casos escribimos  $M(x) = \nearrow$ . El concepto de MT se trata de manera más extensa en [2].

Existen dos tipos de Máquina de Turing a tener en cuenta.



- **Determinista:** Es una Máquina de Turing donde para cualquier estado de la máquina, solo existe una posibilidad o curso de acción para realizar. En otras palabras,  $\delta$  es una función (de transición).
- **No determinista:** Se caracteriza por ser una Máquina de Turing que en cualquier estado de la máquina puede tener más de una (o ninguna) acción a escoger para realizar. Es decir,  $\delta$  no es una función, sino solo una relación.

Podemos decir que una MT determinista solo tiene una opción a realizar en cada paso del proceso, y que si volvemos a correr la máquina en la misma entrada, realizará exactamente los mismos pasos y dará la respuesta obtenida anteriormente; mientras que en una MT no determinista, al haber al menos una posibilidad a efectuar en cada uno de sus pasos, la máquina podría realizar en general procesos distintos, e incluso podría llegar a dar distintas respuestas para la misma entrada.

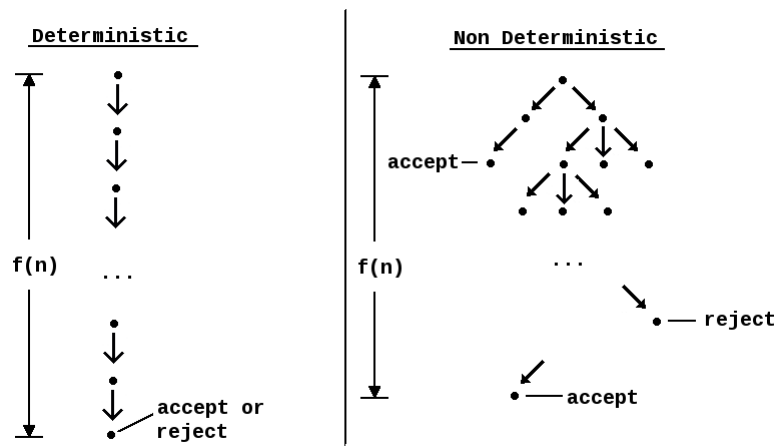


FIGURA 2. Ejemplo de procesos de MT deterministas y no deterministas.

Existe una creencia no demostrable pero con una aceptación casi total, que dice que “todo algoritmo es equivalente a una máquina de Turing”, esto se conoce como la Tesis de Church-Turing. En adelante en el trabajo, cada vez que hagamos mención o mostremos un algoritmo, nos referimos de manera indirecta a una MT.

Incluiremos una pequeña parte sobre teoría de grafos. Se utilizará principalmente en el Capítulo 4 de este trabajo.

**Definición 1.22:** Un *grafo* se define como un conjunto de nodos (vértices) y un conjunto de aristas (lados), donde las aristas permiten unir los nodos. Esto se representa como  $G = \langle V, E \rangle$ ,

donde  $V$  es el conjunto de vértices (numerados) y  $E$  el conjunto de las aristas, donde cada arista se representa como  $(i, j)$ , donde  $i$  indica el nodo inicial y  $j$  el nodo final de la arista.

Se asume que el conjunto de vértices del grafo es un conjunto totalmente ordenado.

Se le puede otorgar dirección a las aristas de un grafo, por ejemplo, en la arista  $(i, j)$ , el nodo  $j$  representa el nodo al cual apunta la arista. Si esto ocurre con todas las aristas del grafo entonces se llama *grafo dirigido*. Para este trabajo consideramos grafos dirigidos que poseen a lo sumo una arista entre dos nodos cualesquiera.

Si se le asocia una valor (peso) a la arista, al grafo se le llama *grafo con aristas de peso*. En el conjunto  $E$  se representan las aristas de peso como la terna  $(i, q, j)$ , donde  $q$  es el peso.

Se tratarán únicamente grafos dirigidos con arista de peso con una cantidad finita de vértices y aristas.

**Definición 1.23:** Definimos la *matriz de adyacencia*, como la matriz obtenida de un grafo dirigido con aristas de peso al aplicar el siguiente proceso:

$$a_{ij} = \begin{cases} q & \text{si } (i, q, j) \in E. \\ 0 & \text{otros casos.} \end{cases}$$

A través de la MT se puede resolver un tipo especial de problema, que se responde con sí o no (únicamente). A esta clase de problema se les denomina *problema de decisión*.

Presentaremos tres clásicos problemas de decisión como ejemplo. El primero es el **problema del camino hamiltoniano (HAMILTON PATH)**: dado un grafo  $G = \langle V, E \rangle$ , averiguar si existe una forma de recorrer el grafo pasando por todos los nodos exactamente una vez. Otro problema asociado a grafos es el **problema de la  $k$ -colorabilidad ( $k$ -COL)**: dado un grafo  $G = \langle V, E \rangle$ , existe alguna forma de colorear los vértices usando solo  $k$  colores de tal forma que cualesquiera dos vértices adyacentes no posean el mismo color.

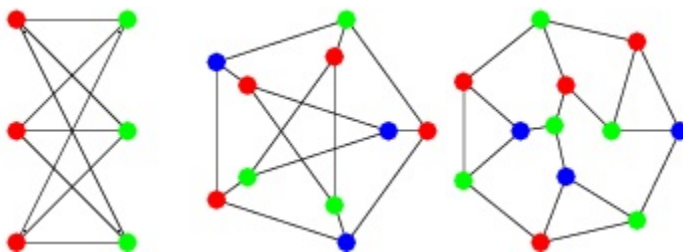


FIGURA 3. Ejemplo de grafos 2 y 3 colorables.

El último problema que mencionaremos en esta sección es el **problema de satisfacibilidad de una fórmula booleana**, también conocido como ( $k$ -SAT), que pregunta si dada una fórmula booleana en forma normal conjuntiva con  $k$  literales por cláusula, existe alguna asignación de verdad para las variables tal que la fórmula se satisfaga.

A continuación mostramos un ejemplo de una instancia de 3-SAT:

$$(x \vee y \vee z) \wedge (x \vee (\neg y) \vee z) \wedge ((\neg x) \vee y \vee z)$$

Ya teniendo lo que es un problema de decisión, es apropiado mencionar que no a todo problema de decisión se le puede hallar una solución, lo cual nos lleva a la siguiente definición.

**Definición 1.24:** Un problema de decisión se dice que es *indecidible* si no existe algoritmo que lo resuelva.

La siguiente pregunta a realizar, ¿Podemos clasificar de manera conveniente todos estos problemas? La respuesta es sí. Vamos a definir conjuntos que nos permitan clasificar una parte de estos problemas (ya que existen problemas que aún se desconoce donde se encuentran en la clasificación).

**Definición 1.25:** Una *clase de complejidad* es un conjunto de problemas de decisión que se pueden resolver con el mismo tipo de Máquina de Turing en un determinado **orden de tiempo** o **espacio**.

Con orden de tiempo o espacio nos referimos; en tiempo, a la cantidad de pasos que debe realizar la MT para poder llegar a la respuesta, mientras que en espacio nos referimos a la cantidad de información que debe almacenar la MT para poder resolver el problema.

Por ejemplo, si decimos que una MT resuelve un problema en tiempo  $f(n)$ , significa que como máximo realizará una cantidad proporcional a  $f(n)$  de pasos para poder llegar a la respuesta (donde  $n$  es el tamaño de la entrada). Para definir el orden de espacio, se calcula con base en la longitud de la cadena final que entrega la MT en términos de  $n$ . Así, si tenemos una MT que trabaja en  $\mathcal{O}(n^2)$ , significa que la salida de la máquina tendrá una longitud proporcional a  $n^2$ , donde  $n$  es la longitud de la entrada.

Teniendo en cuenta esto podemos definir las siguientes clases de complejidad:

**Definición 1.26:** Definimos la *Clase L* como el conjunto de todos los problemas de decisión que se pueden resolver a través de una Máquina de Turing determinista en espacio logarítmico.

**Definición 1.27:** Definimos la *Clase NL* como el conjunto de todos los problemas de decisión que se pueden resolver a través de una Máquina de Turing no determinista en espacio logarítmico.

**Definición 1.28:** Definimos la *Clase P* como el conjunto de todos los problemas de decisión que se pueden resolver a través de una Máquina de Turing determinista en tiempo polinomial.

**Definición 1.29:** Definimos la *Clase NP* como el conjunto de todos los problemas de decisión que se pueden resolver a través de una Máquina de Turing no determinista en tiempo polinomial.

**Definición 1.30:** Definimos la *Clase EXP* como el conjunto de todos los problemas de decisión que se pueden resolver a través de una Máquina de Turing determinista en tiempo exponencial.

Todas estas clases cumplen con la siguiente cadena de contenciones:

$$\mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{EXP}$$

En la actualidad es un problema abierto el hecho de si la contención de  $P$  en  $NP$  es estricta o no. Este es uno de los problemas del milenio y es conocido como el problema de  $P$  vs  $NP$ .

En este punto tenemos a nuestra disposición los problemas de decisión, una forma de resolverlos (a través de una MT) y una clasificación para los mismos. La pregunta natural que uno se puede hacer a continuación es: ¿Existe una forma de relacionar problemas de decisión entre si? La respuesta es sí. Para eso tenemos la próxima definición.

**Definición 1.31:** Dados dos problemas de decisión A y B; definimos una *reducción* del problema A al problema B como un algoritmo que transforma una instancia del problema A en una instancia del problema B en tiempo polinomial o espacio logarítmico.

Es decir, si reducimos una entrada del problema A (llamemos esa entrada  $p$ ) a una entrada del problema B (llamemos esa entrada  $q$ ). La respuesta para  $q$  es sí, si y solo si, la respuesta para  $p$  es sí.

**Observación:** Si podemos reducir el problema A al problema B, al resolver el problema B logramos resolver también el problema A. Esto significa mucho si nos referimos al problema B, ya que si A posee alguna característica especial, como por ejemplo ser indecidible, entonces B tiene que ser indecidible también. Ya que si B se puede resolver, por el hecho de que A se puede reducir a B, A también se puede resolver; lo cual contradice el hecho de que A es indecidible.

Con esta observación queremos mostrar que a través de reducciones podemos heredar características del problema A al problema B. Además, si existe una reducción del problema A al problema B, y el problema B pertenece a la clase de complejidad  $C$ , entonces la clase  $C$  sirve como cota superior para el problema A.

En este trabajo se hará uso de la siguiente notación,  $\mathcal{O}(f(n))$ , que representa un orden de espacio o tiempo  $f(n)$  (como los mostrados en la definición 1.25). Por ejemplo, si tenemos una MT que trabaja en tiempo  $\mathcal{O}(n^2)$ , esto significa que le puede tomar una cantidad de tiempo  $n^2$  o un múltiplo de este tiempo, por ejemplo,  $an^2$ , con  $a \in \mathbb{Z}^+$ , o este tiempo adicionando órdenes de tiempo menores, por ejemplo,  $n^2 + 2n$ , para dar una respuesta en una instancia de tamaño  $n$ .

**Definición 1.32:** Un problema A se dice  $C$ -fuerte si todo problema en C se puede reducir al problema A. Se dice que es  $C$ -completo si es  $C$ -fuerte y además, A pertenece a C.

Los problemas **HAMILTON PATH**, **k-SAT** y **k-COL** (considerando  $k > 2$ ) son NP-completos. La NP-completitud de estos problemas fue demostrada por Richard Karp, usando el Teorema de Cook [6] al realizar una reducción en tiempo polinomial desde el problema de satisfacibilidad booleana a cada uno de los 21 problemas que se mencionan en su trabajo “*Reducibility Among Combinatorial Problems*”(1972).

Existe un tipo particular de MT que nos permite definir una nuevo tipo de algoritmo.

**Definición 1.33:** Un *Algoritmo de Monte Carlo* es un algoritmo no determinista que cumple las siguientes condiciones:

- Si responde SÍ, esta respuesta es correcta con una probabilidad de 100 %.
- Si responde NO, esta respuesta es correcta con una probabilidad de al menos 50 %.

Gracias a este nuevo tipo de algoritmo podemos definir una nueva clase de complejidad.

**Definición 1.34:** La *Clase RP* (Random Polinomyal time) es el conjunto de todos los problemas de decisión que se pueden resolver con un algoritmo de Monte Carlo en tiempo polinomial.

Esta clase será relevante en los Capítulos **3** y **6**.

## Problemas Matriciales

### 1. Introducción

En este trabajo consideraremos la complejidad computacional de algunos problemas de álgebra relacionados con el rango de matrices. Sean  $E$  y  $S$  sub-conjuntos de un anillo conmutativo  $R$  y  $x_1, \dots, x_t$  variables. Dada la matriz  $M = M(x_1, \dots, x_t)$  con entradas escogidas en  $E \cup \{x_1, \dots, x_t\}$ , queremos determinar

$$(2.1) \quad \text{maxrank}_S(M) := \text{máx}\{\text{rank}(M(a_1, \dots, a_t)) : (a_1, \dots, a_t) \in S^t\}$$

y

$$(2.2) \quad \text{minrank}_S(M) := \text{mín}\{\text{rank}(M(a_1, \dots, a_t)) : (a_1, \dots, a_t) \in S^t\}.$$

También hay variantes de estos problemas que especifican más sobre la estructura de  $M$ , o en vez de buscar calcular el rango máximo o mínimo, se pregunta si existe una substitución de las variables que hagan a la matriz invertible o no invertible (nos enfocaremos en estos últimos casos).

Dependiendo de cómo sean  $E$  y  $S$  y cuál sea el problema que estemos estudiando, la complejidad de estos problemas puede pasar de tener solución en tiempo polinomial a tener solución en tiempo aleatorio polinomial, a ser NP-completo o a que no se pueda resolver.

Nuestro marco de trabajo es el siguiente: Si  $E$  es un conjunto no vacío y  $n \in \mathbb{N}$ , entonces  $\mathcal{M}_n(E)$  es el conjunto de la matrices  $n \times n$  con entradas en  $E$ .  $\mathcal{M}(E)$  es la unión de todos los conjuntos  $\mathcal{M}_n(E)$ . Si  $R$  es un anillo conmutativo, entonces  $\mathcal{M}_n(R)$  es el anillo de la matrices  $n \times n$  con entradas en  $R$ . Las filas  $\alpha_i$  de una matriz son linealmente independientes sobre  $R$  si  $\sum c_i \alpha_i = 0$  (con  $c_i \in R$ ) implica que  $c_i = 0$  para todos los  $i$ . Se define de manera similar para las columnas.

El *determinante* de una matriz  $M \in \mathcal{M}_n(R)$  se define como

$$(2.3) \quad \det(M) = \sum_{P \in S_n} [\text{sgn}(P) \cdot a_{1,P(1)} \cdot a_{2,P(2)} \cdots a_{n,P(n)}],$$

donde  $S_n$  es el grupo de permutaciones de orden  $n$ .

El rango de una matriz  $M$ ,  $\text{rank}(M)$ , es el número de filas linealmente independientes. El rango también se puede definir como el número de columnas linealmente independientes. Además, es bien sabido que ambas definiciones coinciden. Una matriz  $M$  de orden  $n \times n$  es invertible, si y solo si,  $\text{rank}(M) = n$ .

Para los problemas de  $\text{maxrank}_S(M)$  y  $\text{minrank}_S(M)$  no pediremos mostrar necesariamente la  $t$ -tupla en la cual  $M$  tenga rango máximo o mínimo.

Una operación que usaremos en este trabajo será la de tomar una lista de matrices cuadradas  $M_1, M_2, \dots, M_k$  y construir una matriz  $M$  colocando a cada  $M_i$  de manera consecutiva en la diagonal principal y ceros en las demás entradas. Esto lo denotaremos por  $M = \text{diag}(M_1, M_2, \dots, M_k)$ . Por propiedades de matrices, tenemos que

$$(2.4) \quad \det(M) = \prod_{1 \leq i \leq k} \det(M_i);$$

$$(2.5) \quad \text{minrank}_S(M) \geq \sum_{i=1}^k \text{minrank}_S(M_i);$$

$$(2.6) \quad \text{maxrank}_S(M) \leq \sum_{i=1}^k \text{maxrank}_S(M_i).$$

Demostremos solamente la desigualdad (2.5), ya que la desigualdad con  $\text{maxrank}$  se resuelve de manera análoga.

**Proposición 2.1:** Sean  $R$  un anillo conmutativo,  $E, S \subseteq R$  y  $M = \text{diag}(M_1, \dots, M_k)$  con  $M_1, \dots, M_k \in \mathcal{M}(E \cup \{x_1, \dots, x_t\})$ , donde  $x_1, \dots, x_t$  son distintas variables. Se satisface que

$$\text{minrank}_S(M) \geq \sum_{i=1}^k \text{minrank}_S(M_i).$$

**Demostración:** Sin pérdida de generalidad, podemos suponer que  $M_i$  no es la matriz nula para todo  $i$ .

Por definición de  $M$ , tenemos que para cualquier tupla  $(a_1, \dots, a_t)$  es cierto que  $\text{rank}(M) = \text{rank}(M_1) + \dots + \text{rank}(M_k)$ .

Sea  $m_i = \text{minrank}_S(M_i)$  para todo  $i$ . Hay dos casos; existe una  $t$ -tupla  $(a_1, \dots, a_t) \in S^t$  tal que cada matriz  $M_i$  alcanza su rango mínimo en  $(a_1, \dots, a_t)$ ; o para cualquier  $t$ -tupla de  $S^t$  por lo menos una de las matrices no alcanzará su rango mínimo.



En el primer caso es fácil ver que  $\text{minrank}_S(M) = m_1 + \dots + m_k$ . En el segundo caso, siempre que evaluemos un elemento de  $S^t$  y calculemos el rango de  $M$ , habrá por lo menos una sección de la matriz  $M$  que no logrará el rango mínimo. Por lo cual, al calcular el mínimo rango de  $M$ , tendremos que  $\text{minrank}_S(M) > m_1 + \dots + m_k$ . Así, se obtiene la desigualdad

$$\text{minrank}_S(M) \geq \sum_{i=1}^k \text{minrank}_S(M_i).$$

**Q. E. D.**

## 2. Algunos ejemplos

Antes de describir los resultados de complejidad, ilustraremos los cálculos de  $\text{minrank}$  y  $\text{maxrank}$  con unos ejemplos. Consideremos la matriz

$$M = \begin{bmatrix} 2 & X_1 & 5 \\ X_2 & 7 & X_3 \\ X_4 & X_4 & 0 \end{bmatrix}$$

En esta matriz, si elegimos la tupla  $(X_1, X_2, X_3, X_4) = (7, 2, 5, 0)$  obtenemos que el rango de  $M$  es 1 (que en este caso es el rango mínimo de  $M$ , es decir,  $\text{minrank}(M) = 1$ ); mientras que si elegimos la tupla  $(X_1, X_2, X_3, X_4) = (1, 1, 1, 1)$  tenemos que el rango de  $M$  es 3, (que en este caso es el rango máximo de  $M$ , es decir,  $\text{maxrank}(M) = 3$ ).

Los valores  $\text{minrank}_S(M)$  y  $\text{maxrank}_S(M)$  pueden depender de quien sea  $S$ . Mostraremos con otro ejemplo este punto. Si consideramos  $S = \mathbb{Z}$  y

$$M = \begin{bmatrix} 3 & 10 \\ X & 7 \end{bmatrix}$$

se tiene que  $\text{minrank}_{\mathbb{Z}}(M) = 2$ , mientras que si consideramos a  $S = \mathbb{Q}$  tenemos que  $\text{minrank}_{\mathbb{Q}}(M) = 1$ . Este es un ejemplo de la siguiente propiedad: Si  $S \subseteq S'$  entonces  $\text{minrank}_S(M) \geq \text{minrank}_{S'}(M)$ , la cual demostraremos más adelante.

Ocurre algo similar con  $\text{maxrank}$ . Si consideramos  $S = GF(2)$ , el cuerpo (finito) con dos elementos y

$$M = \begin{bmatrix} X & X \\ 1 & X \end{bmatrix}$$

en este caso obtenemos  $\text{maxrank}_{GF(2)}(M) = 1$ , mientras que si  $S = GF(4)$  tenemos que  $\text{maxrank}_{GF(4)}(M) = 2$ ; esto lo certificamos tomando  $X \neq 0$  y  $X \neq 1$ . Así, podemos apreciar que: Si  $S \subseteq S'$  entonces  $\text{maxrank}_S(M) \leq \text{maxrank}_{S'}(M)$ . A continuación demostraremos la propiedad antes mencionada para  $\text{minrank}$ .

**Proposición 2.2:** Si  $S \subseteq S'$  entonces  $\text{minrank}_S(M) \geq \text{minrank}_{S'}(M)$ .

**Demostración:** Sean los conjuntos  $S_1, S_2$  tales que  $S_1 \subseteq S_2$  y  $M \in \mathcal{M}_n(E \cup \{x_1, \dots, x_t\})$ . Usando el hecho de que  $S_1 \subseteq S_2$  se tiene que

$$\{\text{rank}(M(a_1, \dots, a_t)) : (a_1, \dots, a_t) \in S_1^t\} \subseteq \{\text{rank}(M(a_1, \dots, a_t)) : (a_1, \dots, a_t) \in S_2^t\}.$$

Luego, por propiedades de mínimo y la Definición 2.2,  $\text{minrank}_{S_1}(M) \geq \text{minrank}_{S_2}(M)$ .

**Q. E. D.**

La propiedad para  $\text{maxrank}$  se demuestra de manera análoga.

### 3. Presentación de los problemas

La mayoría de los resultados de complejidad al momento de computar  $\text{maxrank}$  y  $\text{minrank}$  se pueden escribir de manera natural en términos de los problemas de decisión mostrados en la Tabla 1.

Primero definiremos lo que compone a los problemas.

Fijo:  $R$ , un anillo conmutativo.

$E, S \subseteq R$ .

Entrada:  $M$ , una  $n \times n$  matriz con entradas en  $E \cup \{x_1, \dots, x_t\}$ .

$k$ , un entero no negativo.

Problema	Entrada	Descripción
MINRANK	$M, k$	$\text{¿ } \text{minrank}_S(M) \leq k \text{ ?}$
MAXRANK	$M, k$	$\text{¿ } \text{maxrank}_S(M) \geq k \text{ ?}$
SING	$M$	$\text{¿ } \exists (a_1, \dots, a_t) \in S^t : \det(M(a_1, \dots, a_t)) = 0 \text{ ?}$
NONSING	$M$	$\text{¿ } \exists (a_1, \dots, a_t) \in S^t : \det(M(a_1, \dots, a_t)) \neq 0 \text{ ?}$

TABLA 1. Problemas de decisión.

Los problemas MINRANK y MAXRANK son las versiones de decisión del problema de calcular minrank y maxrank de una matriz cualquiera. Mientras que los problemas SING (por la palabra singularity) y NONSING representan la pregunta:

¿Existe una tupla perteneciente a  $S^t$  que haga a la matriz dada singular (no singular)?

Observando la tabla, podemos notar que los problemas tienen relación entre ellos. Con esto nos referimos al hecho de que si queremos resolver el problema SING para una matriz  $M$  dada, podemos en vez, resolver el problema MINRANK usando la matriz  $M$  y considerando  $k = n$ . Es decir, existe una reducción natural del problema SING al problema MINRANK. Ocurre de manera análoga con los problemas NONSING y MAXRANK, tomando  $k = n$ .

#### 4. Resumen de resultados

A continuación mostraremos una tabla que resume los resultados de complejidad sobre los cuatro problemas de decisión.

$S$	$E$	MAXRANK NONSING	SING	MINRANK
$GF(q)$	$\{0, 1\} \subseteq E \subseteq GF(q)$	NP-completo		
$\mathbb{Z}$	$\{0, 1\} \subseteq E \subseteq \mathbb{Q}$	RP	Indecible	
$\mathbb{Q}$			NP-fuerte	
$\mathbb{R}$				
$\mathbb{C}$				

TABLA 2. Cotas de complejidad para los problemas de decisión.

En la Tabla 2 colocamos los problemas NONSING y MAXRANK juntos, ya que no hemos sido capaces de separar sus complejidades, aunque no sabemos si poseen la misma en general. Pero sí tenemos conocimiento de que los problemas SING y MINRANK no poseen en general la misma complejidad ya que sobre  $\mathbb{C}$  el problema MINRANK es NP-fuerte, mientras que SING se resuelve en tiempo aleatorio polinomial.

Quien sea exactamente  $E$  no es importante para nuestras cotas de complejidad. Todas nuestras cotas inferiores son válidas cuando  $E = \{0, 1\}$ , y todas nuestras cotas superiores son

válidas si  $E$  es  $\mathbb{Q}$  o un cuerpo extensión de dimensión finita de  $\mathbb{Q}$  (respectivamente cuando  $E$  es  $GF(q)$  o un cuerpo extensión de dimensión finita de  $GF(q)$ , cuando la característica es finita). Para las cotas superiores asumimos que el tamaño de la entrada es el número total de bits necesarios para listar por separado cada una de las entradas de la matriz  $M$ , usando la representación binaria estándar para representar los números y representando polinomios como vectores de sus coeficientes. Las cotas superiores son fuertes en otro sentido. Podemos permitir polinomios multivariados (con coeficientes en  $E$ ) en una entrada de la matriz  $M$  y aún preservar las cotas superiores, ya que tal polinomio multivariado puede ser escrito como una fórmula aritmética usando la multiplicación y adición binaria, pero sin hacer uso de exponenciación. Así, la longitud de la representación de un polinomio multivariado es por lo menos tan largo como su grado.

$S$  es importante al momento de considerar la complejidad, como se muestra en la Tabla 2. Sin embargo, las cotas inferiores y superiores para  $S = \mathbb{C}$  en realidad son válidas para cualquier cuerpo algebraicamente cerrado.

Los resultados de la Tabla 2 podemos resumirlos en tres casos: (1) La cota superior de tiempo aleatorio polinomial es obtenida por el artículo de *Schwartz* [3], (2) el resultado de indecidibilidad para  $S = \mathbb{Z}$  en el cual usamos el *Décimo Problema de Hilbert*, (3) todos los demás problemas de la Tabla 2 (que no hayan sido clasificados como RP o indecidibles) son equivalentes bajo reducciones polinomiales a resolver la Teoría Existencial de Primer Orden (TEPO, esto se define en el Capítulo 5) sobre el cuerpo  $S$ . La equivalencia del caso (3) implica que todos estos problemas son NP-fuertes.

También consideraremos el caso especial cuando una variable ocurre exactamente una vez. Ninguna de nuestras demostraciones para las cotas inferiores son válidas bajo esta restricción, además que habría que mejorar algunas de nuestras cotas superiores. En la Tabla 3 se muestran los resultados para este caso particular. Las nuevas cotas superiores se basan en el polinomio del determinante siendo multi-afín. En tal caso el RP-algoritmo para SING sobre  $\mathbb{C}$  puede ser generalizado para demostrar SING sobre cualquier cuerpo.

$S$	$E$	MAXRANK NONSING	SING
$GF(q)$	$GF(q)$	RP	
$\mathbb{Q}$	$\mathbb{Q}$		
$\mathbb{R}$			
$\mathbb{C}$			

TABLA 3. Cotas superiores para los problemas de decisión donde cada variable aparece exactamente una vez.

## Jerarquía de los Problemas

### 1. Computar MAXRANK sobre cuerpos infinitos

En esta sección mostraremos cómo computar MAXRANK con un algoritmo aleatorio (de Monte Carlo específicamente) en tiempo polinomial sobre cualquier cuerpo infinito. También mostraremos que para resolver el problema para  $R = S = \mathbb{F}$ , basta considerar el caso en que  $R = S = \mathbb{Z}$ , cuando  $\mathbb{Z} \subseteq \mathbb{F}$ . Nuestra herramienta principal será una adaptación del siguiente lema de *Schwartz* [3]:

**Lema 3.1:** Sea  $\mathbb{F}$  un cuerpo y sea  $p(x_1, \dots, x_t)$  un polinomio multivariado de grado total a lo sumo  $d$ , el cual no es idénticamente cero. Sea  $V$  cualquier sub-conjunto no vacío de  $\mathbb{F}$ . Entonces en el conjunto  $V^t$ ,  $p$  posee a lo sumo  $d|V|^{t-1}$  ceros.

**Demostración:** Antes de empezar la demostración, re-escribimos el polinomio  $p$  de una manera particular.

Sea  $Q_1$  la forma estándar simplificada de  $p$ , es decir,  $p(x_1, \dots, x_t) = Q_1(x_1, \dots, x_t) = Q_2(x_2, \dots, x_t)x_1^{d_1} + Q_{2(1)}(x_2, \dots, x_t)x_1^{d_1-1} + \dots + Q_{2(d_1-1)}(x_2, \dots, x_t)x_1 + Q_{2(d_1)}(x_2, \dots, x_t)$ ; donde  $d_1$  es el grado de  $x_1$  en  $Q_1$  y  $Q_2$  el coeficiente multivariado que acompaña a  $x_1^{d_1}$ .

Ahora procederemos con la demostración haciendo inducción sobre el número de variables  $t$ . Para el caso  $t = 1$ , tenemos un polinomio  $p(x)$  que no es idénticamente cero y de grado a lo sumo  $d$ . Entonces por el Lema 1.1,  $p$  posee a lo sumo  $d$  ceros, lo cual se puede escribir como  $d = d|V|^0$ .

Ahora lo suponemos cierto para  $t < n$  y lo comprobamos para  $t = n$ . Sea  $p(x_1, \dots, x_n)$  un polinomio que no es idénticamente cero y de grado a lo sumo  $d$ . Escribimos  $p$  en forma estándar simplificada (es decir,  $p = Q_1$ ). Consideraremos los siguientes casos posibles para  $Q_2$ :

- $Q_2$  no tiene raíces. Por lo cual, para cualquier tupla  $(z_2, \dots, z_n) \in V^{n-1}$ , tenemos que  $Q_1(x_1, z_2, \dots, z_n)$  es un polinomio no nulo de una variable y por el Lema 1.1 se tiene que posee a lo sumo  $d_1$  ceros. En este caso, la máxima cantidad de ceros que

puede poseer  $Q_1$  es  $d_1|V|^{n-1}$ , donde  $|V|^{n-1}$  es el número de todas las tuplas de la forma  $(z_2, \dots, z_n)$ .

- $Q_2$  tiene raíces. Como  $Q_2$  es un polinomio de  $n-1$  variables, por hipótesis inductiva tenemos que  $Q_2$  posee a lo sumo  $(d-d_1)|V|^{n-2}$  ceros; además, si  $(z_2, \dots, z_n)$  es una de tales raíces, entonces  $Q_1(x_1, z_2, \dots, z_n)$  puede ser cero para todo  $x_1 \in V$ .

Así, el número total de ceros de  $p$  en  $V^n$  está acotado por

$$\begin{aligned} d_1|V|^{n-1} + (d-d_1)|V|^{n-2} \cdot |V| &= (d_1 + d - d_1)|V|^{n-1} \\ &= d|V|^{n-1}. \end{aligned}$$

### Q. E. D.

Para tener el lema que necesitamos, utilizaremos el que acabamos de demostrar, solo que usaremos  $\mathbb{F}$  y  $V$  con una cierta cantidad de elementos.  $\mathbb{F}$  tendrá al menos  $2d$  elementos y  $V$  tendrá exactamente  $2d$  elementos. Si sustituimos  $d = \frac{|V|}{2}$  en el resultado del lema anterior se obtiene que:

$$d|V|^{t-1} = \frac{1}{2}|V|^t.$$

Así, obtenemos el siguiente resultado.

**Lema 3.2:** Sea  $p(x_1, \dots, x_t)$  un polinomio multivariado de grado total a lo sumo  $d$ , que no es idénticamente cero sobre un cuerpo  $\mathbb{F}$ , el cual contiene al menos  $2d$  elementos. Si  $V$  es cualquier conjunto de  $2d$  elementos de  $\mathbb{F}$ , entonces  $p(\mathbf{a}) \neq 0$  para al menos el 50% de los  $\mathbf{a} \in V^t$ .

**Teorema 3.3:** Sea  $M = M(x_1, \dots, x_t)$  una matriz  $n \times n$  con entradas en  $\mathbb{F} \cup \{x_1, \dots, x_t\}$ . Sea  $V \subseteq \mathbb{F}$  un conjunto finito de al menos  $2n$  elementos. Para cualquier  $t$ -tupla  $(a_1, \dots, a_t) \in V^t$  aleatoria, se tiene con una probabilidad de al menos  $1/2$  que

$$\max_{\mathbf{a} \in V^t} \text{rank}_{\mathbb{F}}(M) = \text{rank}(M(a_1, \dots, a_t)).$$

**Demostración:** Supongamos  $\max_{\mathbf{a} \in V^t} \text{rank}_{\mathbb{F}}(M) = k$ . Entonces existe una  $t$ -tupla  $(a_1, \dots, a_t) \in V^t$  tal que el  $\text{rank}(M(a_1, \dots, a_t)) = k$ . En particular, debe haber una sub-matriz  $k \times k$  de  $M(a_1, \dots, a_t)$  con determinante no nulo. Consideremos la correspondiente sub-matriz  $k \times k$   $M'$  de  $M(x_1, \dots, x_t)$ . Luego el determinante de  $M'$ , considerado como un polinomio multivariado  $p$  en las variables  $x_1, \dots, x_t$ , no puede ser idénticamente cero (ya que es no

nulo cuando  $x_1 = a_1, \dots, x_t = a_t$ ). Ahora, usando el Lema 3.2 (tomando  $d = n$ ),  $p$  es no nulo para al menos la mitad de los elementos de  $V^t$ . Así, para al menos la mitad de las  $t$ -tuplas  $(a_1, \dots, a_t)$ , la correspondiente sub-matriz  $k \times k$  de  $M$  debe ser no singular y por lo tanto  $M(a_1, \dots, a_t)$  tiene rango por lo menos  $k$ . Como  $\text{maxrank}_{\mathbb{F}}(M) = k$ , se tiene que  $\text{rank}(M(a_1, \dots, a_t)) = k$  para al menos la mitad de los elementos de  $(a_1, \dots, a_t) \in V^t$ .

### Q. E. D.

El teorema implica que un algoritmo aleatorio (Apéndice, Pseudo-código 1) en tiempo polinomial puede estimar el  $\text{maxrank}$  de una matriz sobre cualquier cuerpo infinito  $\mathbb{F}$  (es decir,  $\text{MAXRANK} \in RP$ ). Elegimos  $r$   $t$ -tuplas de la forma  $(a_1, \dots, a_t)$  independientemente de manera aleatoria y computamos  $\text{rank}(M(a_1, \dots, a_t))$  para cada una de ellas, obteniendo los rangos  $b_1, \dots, b_r$ . Luego, con una probabilidad de al menos  $1 - 2^{-r}$  tenemos que  $\text{maxrank}_{\mathbb{F}}(M) = \max_{1 \leq i \leq r} b_i$ .

Del Teorema 3.3 se deduce que sobre un cuerpo infinito  $\mathbb{F}$ , la cantidad  $\text{maxrank}(M)$  no cambia si consideramos un cuerpo extensión  $\mathbb{F}'$  con  $\mathbb{F} \subseteq \mathbb{F}'$ , o cuando consideramos un subconjunto infinito  $S \subseteq \mathbb{F}$ . El algoritmo (Pseudo-código 1) funciona exactamente igual siempre y cuando  $V \subseteq S \subseteq \mathbb{F} \subseteq \mathbb{F}'$ . En particular, si  $\mathbb{F}$  tiene característica cero,  $\text{maxrank}_{\mathbb{F}}(M) = \text{maxrank}_{\mathbb{Z}}(M)$ . Más aún, el teorema también implica que el problema de decisión  $\text{MAXRANK}$  pertenece a la clase de complejidad  $RP$  para  $E = \mathbb{Q}$  y  $\mathbb{Z} \subseteq S$ .

## 2. Problema SING sobre un cuerpo algebraicamente cerrado

En esta sección consideramos la complejidad del problema de decisión  $\text{SING}$  en el caso  $R = S = \mathbb{F}$ , donde  $\mathbb{F}$  es un cuerpo algebraicamente cerrado. Mostraremos que en este caso,  $\text{SING} \in RP$ . Primero probaremos los siguientes dos lemas

**Lema 3.4:** Sea  $p(x_1, \dots, x_t)$  un polinomio multivariado sobre un cuerpo infinito  $\mathbb{F}$ .  $p$  es idénticamente cero, si y solo si,  $p$  es el polinomio cero.

**Demostración:** Si  $p$  es el polinomio cero es evidente que es idénticamente cero.

Para demostrar el otro sentido del bicondicional emplearemos una prueba por contrarrecíproco. Supongamos que  $p$  no es el polinomio cero y realicemos inducción en  $t$ , el número de variables. Si  $t = 1$ , entonces  $p$  es un polinomio de una variable de grado  $d$ , con  $d \geq 1$ . Por el Lema 1.1,  $p$  tiene a lo sumo  $d$  raíces, y como  $\mathbb{F}$  es infinito,  $p(a) \neq 0$  excepto para una cantidad finita de  $a \in \mathbb{F}$ .



Ahora suponemos que es cierto para todo  $t < k$  y probamos el caso  $t = k$ . Sea  $p(x_1, \dots, x_k)$  un polinomio multivariado. Sin pérdida de generalidad, elegimos la variable  $x_1$  en  $p$ , que posee grado digamos  $d$ , y reescribimos  $p$  como un polinomio en función de  $x_1$  con coeficientes multivariados, es decir,  $p = z_d x_1^d + \dots + z_1 x_1 + z_0$  (si se encuentra escrito de esta forma se dice que está en forma estándar simplificada). Como  $p$  es no constante, tenemos que  $d \geq 1$  y además que  $z_d$  es un polinomio en  $k - 1$  variables que no es el polinomio cero. Por hipótesis inductiva,  $z_d$  no es idénticamente cero. Escogemos  $(a_2, \dots, a_k) \in \mathbb{F}^{k-1}$  tal que  $z_d \neq 0$ , y llamamos al polinomio bajo esta selección como  $q(x_1) = p(x_1, a_2, \dots, a_k)$ . Luego,  $q$  no es el polinomio cero y por hipótesis inductiva, no es idénticamente cero.

**Q. E. D.**

**Lema 3.5:** Sea  $p(x_1, \dots, x_t)$  un polinomio multivariado no constante sobre un cuerpo  $\mathbb{F}$ . Si  $\mathbb{F}$  es algebraicamente cerrado entonces  $p$  toma todos los valores de  $\mathbb{F}$ .

**Demostración:** Probaremos el resultado a través de inducción en  $t$ , la cantidad de variables. Si  $t = 1$ , entonces  $p = p(x)$  es un polinomio no constante de una variable. Para mostrar que  $p$  toma todos los valores de  $\mathbb{F}$  (o equivalentemente, que el rango de  $p$  es  $\mathbb{F}$ ), consideramos la ecuación  $p(x) - c = 0$  con  $c \in \mathbb{F}$ . Como  $\mathbb{F}$  es algebraicamente cerrado entonces existe  $x = x_0$  tal que es solución de la ecuación; de donde  $p(x_0) = c$ . Como  $c$  es arbitrario, se tiene lo deseado.

Supongamos cierto el caso  $t < k$ . Ahora comprobemos que se cumple para el caso  $t = k$ . Escribimos  $p(x_1, \dots, x_k) = y_1 + y_2 + \dots + y_r$ , donde cada  $y_i$  (que podría ser constante) es un monomio de la forma  $a_i x_1^{e_{i1}} x_2^{e_{i2}} \dots x_k^{e_{ik}}$ . Más aún, supongamos que todos los coeficientes están *reunidos*, i.e., nunca pasará al mismo tiempo que

$$(3.1) \quad i \neq j \quad y \quad (e_{i1}, e_{i2}, \dots, e_{ik}) = (e_{j1}, e_{j2}, \dots, e_{jk})$$

Sin pérdida de generalidad, elegimos la variable  $x_1$  que ocurre con grado  $e$  en  $p$ ; en la forma de  $x_1^e$ . Como  $p$  es no constante y  $x_1$  aparece, entonces  $e \geq 1$ . Ahora, tratamos a  $p$  como un polinomio en función de  $x_1$  con coeficientes multivariados, y lo escribimos en forma estándar simplificada,  $p = z_e x_1^e + \dots + z_1 x_1 + z_0$ , donde cada  $z_i$  es un polinomio de  $k - 1$  variables. Veamos que  $z_e$  no es el polinomio cero. Para ver eso suponemos  $z_e = 0$ .

No tiene sentido pensar que  $z_e$  está conformado por un único  $y_i$ , ya que si esto pasa directamente  $y_i = 0$  y eso no tendría sentido respecto a la escritura de  $p$ ; así que podemos

asumir (sin pérdida de generalidad) que  $z_e$  es obtenido al sacar factor común  $x^e$  de  $y_1$  y  $y_2$ . Como  $z_e = 0$  entonces  $y_1 = -y_2$ , por lo cual el exponente de  $x_i$  para todo  $i$ , es igual en  $y_1$  y  $y_2$ ; lo cual contradice la propiedad 3.1. Por lo tanto,  $z_e$  no es el polinomio cero y por el Lema 3.4, existe  $(a_2, \dots, a_k) \in \mathbb{F}^{k-1}$  tal que  $z_e \neq 0$ . Usando esta asignación para todas las variables en  $p$  obtenemos un polinomio no constante en función de  $x_1$ . Luego, por hipótesis inductiva obtenemos el resultado deseado.

### Q. E. D.

Gracias a estos dos lemas podemos demostrar el segundo resultado de interés de esta sección, el cual mostraremos como el siguiente teorema.

**Teorema 3.6:** Si  $S = R = \mathbb{F}$ , y  $\mathbb{F}$  es un cuerpo algebraicamente cerrado, entonces  $\text{SING} \in RP$ .

**Demostración:** Sea  $M(x_1, \dots, x_t)$  una matriz. Consideremos el siguiente algoritmo: Sea  $V \subseteq \mathbb{F}$  de al menos  $2n$  elementos distintos, donde  $n$  es el orden de la matriz  $M$ . Definimos el polinomio  $p(x_1, \dots, x_t) = \det(M(x_1, \dots, x_t))$ . Elegimos  $r$   $t$ -tuplas  $\mathbf{a}_1, \dots, \mathbf{a}_r$  aleatoriamente de  $V^t$  y evaluamos el determinante de  $M$  en  $\mathbf{a}_i$ ,  $\det(M(\mathbf{a}_i))$ , para  $1 \leq i \leq r$ . Se nos presentarán tres posibles casos al hacer esto.

- Si de los  $r$  resultados obtenemos al menos dos distintos, entonces respondemos sí (debido al Lema 3.5,  $p$  toma el valor 0).
- Si todos los valores son iguales y son cero entonces respondemos sí.
- Si todos los valores son iguales y son no nulos entonces respondemos no.

El algoritmo siempre da la respuesta correcta a menos que hayamos obtenido todos los valores iguales y no nulos. En este caso respondemos no, cuando la respuesta correcta podría ser sí (Falso negativo). Si el polinomio no es constante, entonces el polinomio  $p - p(\mathbf{a}_1)$  no es el polinomio nulo y por el Lema 3.2 sabemos que por lo menos con un 50% de probabilidad,  $p(\mathbf{a}_i) \neq p(\mathbf{a}_1)$  para  $2 \leq i \leq r$ . Luego, la probabilidad de cometer un error está acotada superiormente por  $1/2^{r-1}$ , o la probabilidad de que el “no” obtenido sea la respuesta correcta está acotada inferiormente por  $1 - 1/2^{r-1}$ .

### Q. E. D.

Haciendo solo unos pequeños cambios a esta demostración podemos obtener el resultado  $\text{NONSING} \in RP$ . Solo tenemos que cambiar la manera como el algoritmo responde después de haber evaluado las  $r$   $t$ -tuplas:

- Si de los  $r$  resultados obtenemos al menos dos distintos, entonces respondemos sí (ya que obtenemos un valor distinto de cero).
- Si todos los valores son iguales y son no nulos entonces respondemos sí.
- Si todos los valores son iguales y son cero entonces respondemos no.

También para este caso tenemos que la probabilidad de que el “no” obtenido sea la respuesta correcta esta acotada inferiormente por  $1 - 1/2^{r-1}$  (debido al Lema 3.2).

En el apéndice, Pseudo-código 2, se muestra el algoritmo que respalda al Teorema 3.6.

## Indecibilidad a través del determinante

### 1. Universalidad del determinante

En este capítulo demostraremos un resultado que se encuentra presente en todas nuestras cotas inferiores para los problemas SING y MINRANK, y que dice que cualquier polinomio multivariado es el determinante de una matriz lo *suficientemente pequeña*. Para la siguiente demostración introduciremos nuevas definiciones. Primero, una *fórmula aritmética* es una fórmula que se encuentra conformada solo por constantes, variables, el operador unario  $\{-\}$  y los operadores binarios  $\{+, \cdot\}$ . Definimos la *longitud* de una fórmula  $F$  (denotada por  $|F|$ ) como el número total de apariciones de constantes, variables y operadores. Por ejemplo:

$$(4.1) \quad |5x - 42y + 3xz| = |5 \cdot x + (-42 \cdot y) + 3 \cdot x \cdot z| = 14$$

También definiremos las siguientes dos cantidades:  $S =$  **número de símbolos + en  $F$** ; y  $F' = |F| - S$ , donde  $F'$  representa la cantidad de símbolos en  $F$  sin tomar en cuenta los símbolos  $+$ . Para el ejemplo anterior tenemos  $S = 2$  y  $F' = 12$ .

En [1] definen longitud de una fórmula como hemos ilustrado en (4.1), pero aprovecharemos para notar que hemos agregado nuevos términos ( $S$  y  $F'$ ), los cuales nos permitirán obtener una medida de longitud de  $F$  alternativa y que para fines de la próxima proposición, nos será más útil y con resultados más precisos, que los obtenidos en [1].

**Observación:** Es distinto referirnos a la fórmula  $F$  que a  $p_F$ , donde  $p_F$  denota al polinomio descrito por la fórmula  $F$ , ya que  $F$  es una cadena de caracteres, a la cual no podemos aplicar ninguna operación aritmética, mientras que a  $p_F$  sí, ya sea a los términos que lo componen o a él mismo como polinomio.

**Proposición 4.1:** Sea  $R$  un anillo conmutativo y  $F$  una fórmula aritmética cuyas constantes provienen de  $E \subseteq R$  y sus variables son  $\{x_1, \dots, x_t\}$ . Para  $n = F' + 2$ , se puede construir en tiempo  $n^{\mathcal{O}(1)}$  una matriz  $M$  de tamaño  $n \times n$  con entradas en  $E \cup \{0, 1\} \cup \{x_1, \dots, x_t\}$

tal que  $p_F = \det(M)$  y  $\text{minrank}_R(M) \geq n - 1$ , donde  $p_F$  denota al polinomio descrito por la fórmula  $F$ .

**Demostación:** Para esta demostración primero mostraremos el proceso (algoritmo) de pasar de una fórmula  $F$  a una matriz  $M$ . Luego iremos demostrando uno a uno las propiedades e igualdades requeridas.

Algunas definiciones que usaremos en el algoritmo serán *camino entre nodos y matriz de adyacencia*, definidas como es usual. El algoritmo consiste en construir dos grafos  $G_F$  y  $G'_F$  a partir de la fórmula  $F$ . El grafo  $G_F$  lo usaremos como base para poder construir el grafo  $G'_F$ , por último calculamos la matriz de adyacencia del grafo  $G'_F$ , dicha matriz es el objetivo final del algoritmo. El algoritmo que usaremos es el descrito a continuación:

Sea  $G = \langle V = \{s, t\}, E = \emptyset \rangle$ . En adelante usaremos un símbolo especial  $\%$  que registra los símbolos de  $F$  ya interpretados en el grafo  $G$ . La siguiente construcción se hace en la entrada  $(F, s, t)$ .

- (1) Si  $F = x$  o  $F = c$ , entonces redefinimos  $V$  como  $V \cup \{t_1\}$ ,  $E$  como  $E \cup \{(s, F, t_1), (t_1, 1, t)\}$  y  $F = \%$ , donde  $t_1$  es un nodo nuevo.
- (2) Si  $F = -F_1$ , redefinimos  $V$  como  $V \cup \{t_1\}$ ,  $E$  como  $E \cup \{(t_1, 1, t)\}$  y  $F = \%F_1$ , donde  $t_1$  es un nodo nuevo. Luego ejecutamos los pasos (1-4) en  $(F_1, s, t_1)$ .
- (3) Si  $F = F_1 \cdot F_2$ , redefinimos  $V$  como  $V \cup \{t_1\}$ , y  $F = F_1 \%F_2$ , donde  $t_1$  es un nodo nuevo. Luego ejecutamos los pasos (1-4) en  $(F_1, s, t_1)$  y después en  $(F_2, t_1, t)$ .
- (4) Si  $F = F_1 + F_2$ , redefinimos  $F = F_1 \%F_2$ . Luego ejecutamos los pasos (1-4) en  $(F_1, s, t)$  y después en  $(F_2, s, t)$ .
- (5) Detenemos el proceso cuando todos los símbolos de  $F$  sean  $\%$  o paréntesis.
- (6) Redefinimos  $E$  como  $E \cup \{(t_1, 1, t_1) : t_1 \neq s\}$ .
- (7) Redefinimos  $E$  como  $E \cup \{(t, 1, s)\}$ .
- (8) Calculamos la matriz de adyacencia del grafo  $G$ .

En el apéndice, Pseudo-código 3 se muestra el proceso anterior.

Nótese que en los pasos (1-3) son los únicos en los cuales se agrega un nodo (quitando los dos colocados inicialmente), es decir, por cada símbolo que se encuentra en  $F$  (esto no incluye al símbolo  $+$ ) agregamos un nodo al conjunto  $V$ . Por lo tanto, al finalizar el algoritmo, la cantidad de nodos (que a su vez es el tamaño de la matriz resultante) es  $|V| = n = F' + 2$ .

Al grafo que obtenemos al encontrarnos en el paso (5) del algoritmo lo nombramos como  $G_F$ , al grafo del paso (7) lo nombramos  $G'_F$  y a la matriz obtenido en el paso (8) la nombramos simplemente  $M$ .

En la Figura 4 se muestra en forma de tabla el algoritmo para construir el grafo. Después de la demostración se muestra un ejemplo para obtener la matriz  $M$  a partir de una fórmula  $F$ .

Fórmula $F$	Serie paralela $s$ - $t$ del grafo $G_F$ con aristas de peso
Constante $c$	
Variable $x$	
$F = -F_1$	
$F = F_1 \cdot F_2$	
$F = F_1 + F_2$	

FIGURA 4. Construcción inductiva de  $G_F$ .

Ahora mostraremos que  $\det(M) = p_F$ . Para lograr esto primero construiremos el grafo dirigido  $G_F$  con aristas de peso provenientes del conjunto  $E \cup \{1\} \cup \{x_1, \dots, x_t\}$ , aplicando a  $F$  los pasos de la Figura 4. A tal grafo  $G_F$ , le asociamos el polinomio

$$(4.2) \quad p(G_F) = \sum_{\substack{\pi \text{ un } s-t \\ \text{camino en } G_F}} \left[ (-1)^{\text{long}(\pi)} \cdot \prod_{\substack{e \text{ una} \\ \text{arista de } \pi}} \text{peso}(e) \right],$$

donde  $\text{long}(\pi) =$  cantidad de aristas en  $\pi$ ; y  $\text{peso}(e) =$  valor asignado a la arista.

Por comodidad, simplificaremos expresiones de la fórmula anterior. La expresión:  **$\pi$  un  $s - t$  camino en  $G_F$**  la simplificamos a  **$\pi s \sim t G_F$** ; la expresión:  **$e$  una arista de  $\pi$**  la simplificamos a  **$e \triangleright \pi$** .

Demostraremos por inducción la igualdad  $p_F = p(G_F)$ . Para los casos básicos:

- Suponiendo  $p_F = c$ , entonces  $p(G_F) = (-1)^2 \cdot (1 \cdot c) = c$ .
- Suponiendo  $p_F = x$ , entonces  $p(G_F) = (-1)^2 \cdot (1 \cdot x) = x$ .

Para los siguientes tres pasos inductivos estableceremos ciertas características para los grafos asociados  $G_{F_1}$  y  $G_{F_2}$  (Figura 4).

Para el grafo  $G_{F_1}$ , asumimos que posee  $n_1$  caminos; donde cada camino tiene longitud  $k_i$ ; con  $i \in \{1, \dots, n_1\}$ . Representaremos los caminos de  $G_{F_1}$  con  $\phi$  ( $k_i = \text{long}(\phi_i)$ ).

Para el grafo  $G_{F_2}$ , asumimos que posee  $n_2$  caminos; donde cada camino tiene longitud  $q_j$ ; con  $j \in \{1, \dots, n_2\}$ . Representaremos los caminos de  $G_{F_2}$  con  $\psi$  ( $q_j = \text{long}(\psi_j)$ ).

- Suponiendo que  $p_F = -p_{F_1}$  y que se satisface que  $p_{F_1} = p(G_{F_1})$ , entonces

$$\begin{aligned} p(G_F) &= \sum_{\pi s \sim t G_F} \left[ (-1)^{\text{long}(\pi)} \cdot \prod_{e \triangleright \pi} \text{peso}(e) \right] \\ &= \sum_{i=1}^{n_1} \left[ (-1)^{k_i+1} \cdot (1 \cdot \prod_{e \triangleright \phi_i} \text{peso}(e)) \right] \\ &= (-1) \sum_{i=1}^{n_1} \left[ (-1)^{k_i} \cdot \prod_{e \triangleright \phi_i} \text{peso}(e) \right] \\ &= (-1) \cdot p(G_{F_1}) \\ &= -p(G_{F_1}) \\ &\stackrel{\text{H.I.}}{=} -p_{F_1} \\ &= p_F \end{aligned}$$

- Suponemos  $p_F = p_{F_1} \cdot p_{F_2}$  y se satisface que  $p_{F_1} = p(G_{F_1})$  y  $p_{F_2} = p(G_{F_2})$ ; entonces

$$\begin{aligned}
p(G_F) &= \sum_{\pi \ s \sim t \ G_F} \left[ (-1)^{\text{long}(\pi)} \cdot \prod_{e \triangleright \pi} \text{peso}(e) \right] \\
&= \sum_{i=1}^{n_1} \left[ \sum_{j=1}^{n_2} \left[ (-1)^{k_i+q_j} \cdot \prod_{e \triangleright \phi_i, \psi_j} \text{peso}(e) \right] \right] \\
&= \sum_{i=1}^{n_1} \left[ (-1)^{k_i} \cdot \prod_{e \triangleright \phi_i} \text{peso}(e) \cdot \left[ \sum_{j=1}^{n_2} \left[ (-1)^{q_j} \cdot \prod_{e \triangleright \psi_j} \text{peso}(e) \right] \right] \right] \\
&= \left[ \sum_{i=1}^{n_1} (-1)^{k_i} \cdot \prod_{e \triangleright \phi_i} \text{peso}(e) \right] \cdot \left[ \sum_{j=1}^{n_2} (-1)^{q_j} \cdot \prod_{e \triangleright \psi_j} \text{peso}(e) \right] \\
&= p(G_{F_1}) \cdot p(G_{F_2}) \\
&\stackrel{\text{H.I.}}{=} p_{F_1} \cdot p_{F_2}
\end{aligned}$$

- Suponemos  $p_F = p_{F_1} + p_{F_2}$ , y se satisface que  $p_{F_1} = p(G_{F_1})$  y  $p_{F_2} = p(G_{F_2})$ ; entonces

$$\begin{aligned}
p(G_F) &= \sum_{\pi \ s \sim t \ G_F} \left[ (-1)^{\text{long}(\pi)} \cdot \prod_{e \triangleright \pi} \text{peso}(e) \right] \\
&= \left[ \sum_{i=1}^{n_1} (-1)^{k_i} \cdot \prod_{e \triangleright \phi_i} \text{peso}(e) \right] + \left[ \sum_{j=1}^{n_2} (-1)^{q_j} \cdot \prod_{e \triangleright \psi_j} \text{peso}(e) \right] \\
&= p(G_{F_1}) + p(G_{F_2}) \\
&\stackrel{\text{H.I.}}{=} p_{F_1} + p_{F_2}
\end{aligned}$$

Con esto probamos que  $p_F = p(G_F)$ . Ahora demostraremos la igualdad  $\det(M) = p(G_F)$ .

Para esto buscaremos demostrar que

$$p(G_F) = \sum_{\pi \in G_F} \left[ (-1)^{\text{long}(\pi)} \cdot \prod_{e \in \pi} \text{peso}(e) \right] = \sum_{\sigma \in S_n} \left[ \text{sgn}(\sigma) \cdot \prod_{i=1}^n a_{i, \sigma(i)} \right] = \det(M)$$

Usando el grafo  $G'_F$ , creamos su matriz de adyacencia  $M$ . Asignamos a cada camino  $s - t$  de  $G'_F$  una permutación de la siguiente manera (que mostraremos con un ejemplo):

Consideramos  $G'_F$  un grafo de 6 nodos. Si un camino está conformado por el conjunto de nodos  $\{s = 1, 4, 5, 6 = t\}$  entonces a este camino le asignamos la permutación  $(1 \ 4 \ 5 \ 6)(2)(3)$ .

Así con cada camino  $s - t$  en  $G'_F$ .



Ahora veremos que para cualquier permutación que no sea de esta forma, el coeficiente asociado en el determinante de la matriz tiene que ser cero.

Si consideramos las permutaciones que mapean un nodo a otro nodo que no están relacionados en el grafo, tendremos que a estas relaciones se le asigna el valor cero (como ejemplo tenemos  $a_{1n} = 0$ ).

De esta forma tenemos que cualquier permutación que no esté asociada a un camino del grafo tendrá representación cero en el determinante. Por último, mostraremos con un ejemplo la parte de la ecuación que falta demostrar que es

$$(-1)^{\text{long}(\pi)} = \text{sgn}(\sigma).$$

Si consideramos la permutación  $\sigma = (1\ 4\ 5\ 6)(2)(3)$  esta se puede re-escribir por transposiciones como  $(1\ 4)(4\ 5)(5\ 6)(2)(3)$ , por lo que  $\text{sgn}(\sigma) = -1$ , pero la cantidad de transposiciones son las mismas que el número de aristas (es más, cada transposición representa una arista del camino), por lo tanto  $(-1)^{\text{long}(\pi)} = (-1)^3 = (-1) = \text{sgn}(\sigma)$ . Para demostrar el caso general, consideramos a  $\sigma$  escrito como un producto de transposiciones. Tenemos que  $\text{sgn}(\sigma) = (-1)^\lambda$ , donde  $\lambda =$  número de transposiciones que componen a  $\sigma$ , y además, cada transposición representará una arista de dicho camino. Así, se obtiene  $(-1)^{\text{long}(\pi)} = \text{sgn}(\sigma)$ .

Así, tenemos que  $\det(M) = p_F$ . Ahora, solo nos queda mostrar que  $\text{minrank}_R(M) \geq n-1$ . Demostraremos que  $\det(M') = 1$ , donde  $M'$  es la matriz  $M$  quitando la primera fila y columna, es decir,  $M'$  es una sub-matriz de  $M$  de tamaño  $(n-1) \times (n-1)$ .

Por construcción de  $M$ , podemos asegurar dos cosas sobre  $M'$ :

- 1 La diagonal principal de  $M'$  posee 1 en todas las entradas.
- 2 La fila  $n-1$  de  $M'$  está compuesta solamente por ceros, con excepción de la entrada asociada a la diagonal principal (que es 1).

Buscamos reducir la matriz  $M'$  a una matriz triangular inferior. Usando la fila  $n-1$ , podemos anular todas las entradas de la forma  $m_{i(n-1)}$  de  $M'$  con  $1 \leq i \leq n-2$ . De donde la fila  $n-2$  se queda con todas las entradas a la derecha de la diagonal principal como cero. Con esto, podemos anular las entradas de la forma  $m_{i(n-2)}$  de  $M'$  con  $1 \leq i \leq n-3$ . Repitiendo este proceso con todas las columnas de la matriz  $M'$  obtenemos una nueva matriz  $M''$  que es triangular inferior. Al momento de reducir  $M'$  a  $M''$ , se genera la pregunta: ¿Si un elemento de la diagonal principal puede anularse también durante este proceso? Esto no puede pasar,

ya que para que ocurra tienen que existir entradas  $m_{ij} = 1$  y  $m_{ji} = 1$ ; por lo tanto existe una arista dirigida del nodo  $i$  al  $j$  y viceversa, pero por construcción del grafo  $G_F$  esto no puede ocurrir. Así que no es posible que se modifique (o anule) un elemento de la diagonal principal de  $M'$ .

Luego,  $\det(M') = \det(M'')$ . Por propiedad de matrices triangulares tenemos que  $\det(M'')$  es igual al producto de los términos de su diagonal principal, y como todos son 1, entonces  $\det(M') = \det(M'') = 1$ .

Como  $M'$  tiene determinante constante no nulo y es una sub-matriz de  $M$  tenemos que  $\text{minrank}_R(M) \geq n - 1$ .

### Q. E. D.

El último paso de esta demostración se ha realizado diferente a como se ha hecho en [1]. En [1] han usado un razonamiento combinatorio tomando en cuenta los ciclos y los lazos que componen al grafo  $G'_F$ ; en cambio, en este trabajo se usó un proceso algebraico en donde se simplificó la matriz  $M'$  convirtiéndola en una matriz triangular inferior.

A continuación se encuentra un ejemplo de la proposición anterior. Donde se muestra una fórmula  $F$ , cada uno de los pasos para la construcción de sus grafos  $G_F$  y  $G'_F$  asociados; y la matriz  $M$  tal que  $\det(M) = F$ .

Para este ejemplo calculamos primero  $n$  (la cantidad de nodos) y creamos el nodo 1 y el nodo  $n$ . Luego vamos anexando cada nodo y arista necesaria.

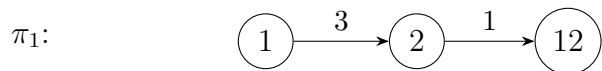
#### Ejemplo 4.2: Algoritmo Proposición 4.1

$$F = 3 \cdot x_1 \cdot x_2 - (x_3 + 2 \cdot x_4)$$

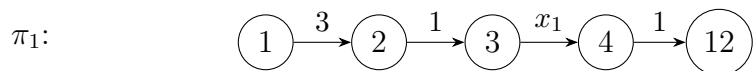
Para la fórmula  $F$  definida, calculamos la cantidad de nodos, que es  $n = 12$ ; luego creamos cada camino del grafo  $G_F$ . Este ejemplo posee 3 caminos, que nombraremos como  $\pi_1$ ,  $\pi_2$  y  $\pi_3$ . Explicaremos paso a paso el proceso para obtener  $\pi_1$ , el camino asociado al término  $3x_1x_2$ . Paso 1, creamos el nodo inicial y el nodo final.

$$\pi_1: \quad \textcircled{1} \quad \textcircled{12}$$

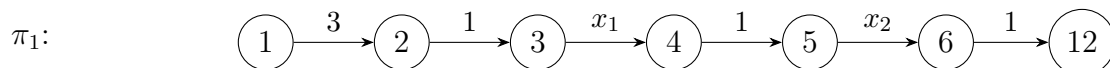
Paso 2, incluimos el nodo y las aristas asociado a la constante 3.



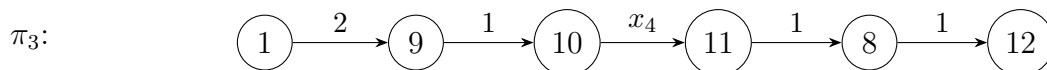
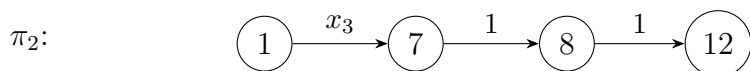
Paso 3, incluimos el nodo y las aristas asociado a la variable  $x_1$ .



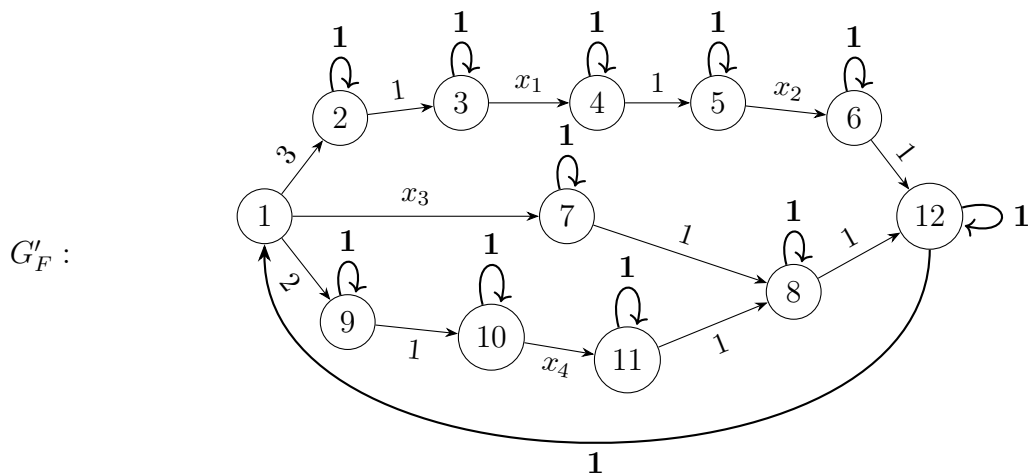
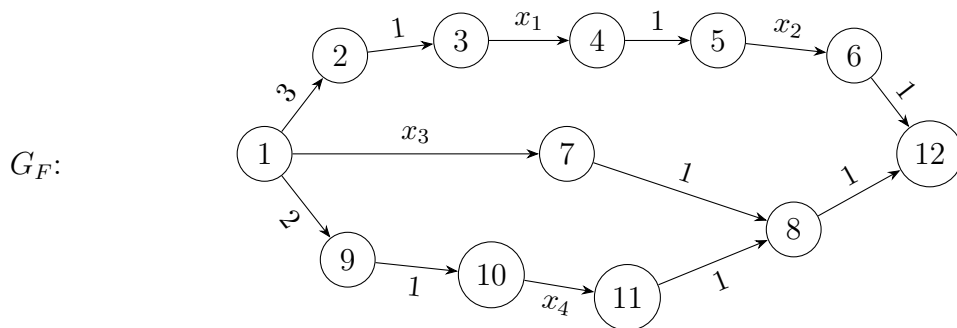
Paso 4, incluimos el nodo y las aristas asociado a la variable  $x_2$ .



Así, obtenemos el primer camino del grafo. Aplicando este proceso al resto de los caminos pertenecientes a  $F$  obtenemos:



Por último, unimos los 3 caminos conectándolos cada uno por el nodo 1 y el nodo 12 respectivamente.



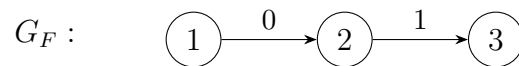
$$M = \begin{bmatrix} 0 & 3 & 0 & 0 & 0 & 0 & x_3 & 0 & 2 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & x_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\det(M) = F = 3 \cdot x_1 \cdot x_2 - (x_3 + 2 \cdot x_4)$$

En el ejemplo anterior mostramos como son los distintos grafos y la matriz construida por el algoritmo para el ejemplo  $F = 3 \cdot x_1 \cdot x_2 - (x_3 + 2 \cdot x_4)$ .

La Proposición 4.1 a pesar de lo extensa que es, se vuelve sencilla de aplicar por la forma procedural de la demostración. Además, que si a primera vista no puede apreciarse su utilidad, la mostraremos más adelante; ya que esta proposición nos permite hallar para cualquier fórmula aritmética una matriz de tamaño lo suficientemente pequeño (hay que considerar también que la mayoría de las entradas de la matriz son ceros.) tal que su determinante sea igual a la fórmula aritmética.

Nos gustaría ilustrar unos casos particulares de la proposición. Uno de ellos es la fórmula  $F = 0$ , la cual, por estar conformada por solo un símbolo de constante, tiene  $n = 3$ .  $F$  posee los siguientes grafos:



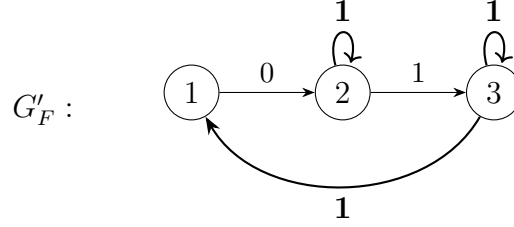


FIGURA 5. Grafos asociados a  $F = 0$ .

De donde se le asocia al grafo  $G'_F$  la matriz de adyacencia  $M$  de abajo. En este caso es fácil ver que  $\text{minrank}(M) = 3 - 1 = 2$ .

$$M = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

Ya ilustrado el caso más sencillo, veremos la propiedad de  $\text{minrank}_R(M) \geq n - 1$ . Solo existen dos opciones:  $\text{minrank}_R(M) = n$  o  $\text{minrank}_R(M) = n - 1$ . Hay una forma de saber, dependiendo de  $p_F$ , en cual de los dos casos se va a encontrar. Si existe una substitución de las variables de  $p_F$  tal que dicha substitución sea una raíz de  $p_F$ , entonces  $\text{minrank}_R(M) = n - 1$ . Si dicha substitución no existe, entonces  $\text{minrank}_R(M) = n$ .

Para probar esto mostraremos primero otro resultado: es posible reducir  $M$  a una matriz triangular inferior, que nombraremos  $M''$ , tal que  $m''_{1,1} = p_F$ . Esto puede probarse al usar parte de la demostración de la Proposición 4.1, en específico la parte de  $\det(M') = 1$ , donde usando los 1 en la diagonal principal podemos anular los elementos que se encuentran encima de ella. Al realizar esto solo usaremos la operación elemental de sumar a una fila un múltiplo de otra fila, para que así  $\det(M'') = \det(M) = p_F$ .

Como  $M''$  es una matriz triangular inferior su determinante es el resultado del producto de los elementos de su diagonal principal; además, por la construcción de  $M$  sabemos que los elementos de la diagonal principal de  $M''$  (exceptuando la entrada  $m''_{1,1}$ ) son todos 1. De esta forma obtenemos que  $\det(M'') = m''_{1,1} = p_F$ .

Con esto demostrado, tenemos que  $\text{minrank}_R(M'') = \text{minrank}_R(M) = n - 1$  cuando  $p_F$  posea una raíz en  $R$ , en caso contrario tenemos  $\text{minrank}_R(M) = n$ .

En los ejemplos podemos ver que la matriz  $M$ , en general, siempre posee una gran cantidad de ceros. Mostraremos como podemos conocer la cantidad de ceros que tendrá la matriz  $M$  usando solo los datos de la fórmula  $F$ , sin calcular la matriz  $M$ . Antes de definir las cantidades que usaremos para el cálculo, demostraremos como obtener dicho valor, el cual denotaremos como  $M_0$  (cantidad de entradas nulas en  $M$ ).

Al momento de crear la matriz de adyacencia, solo las aristas del grafo  $G'_F$  tendrán asociadas entradas no nulas en la matriz.

Fijándonos en el algoritmo de la Proposición 4.1, podemos ver que solo cuando nos encontramos alguno de los símbolos de constante ( $c$ ), variable ( $x$ ) o el signo menos ( $-$ ), se agrega una arista en el grafo. Más aún, cuando encontramos una constante o variable se agregan exactamente dos aristas en vez de una. Si alguno de los símbolos de constante llega a ser cero ( $c = 0$ ), entonces de las dos entradas de la matriz  $M$  asociadas a esas aristas solo una será no nula (para ser exactos, será 1).

También tenemos que por cada nodo (quitando el nodo  $s = 1$ ) existe una arista de el nodo a sí mismo, es decir,  $n - 1$  lazos; si incluimos la arista que va del nodo  $t = n$  al nodo  $s = 1$  obtenemos  $n$  aristas más en el grafo. Con esto dicho definimos:

- $cv$  = cantidad de símbolos de constante y de variable en  $F$ .
- $r$  = cantidad de signos menos ( $-$ ) en  $F$ .
- $F_0$  = cantidad de constantes 0 ( $c = 0$ ) en  $F$ .
- $u$  = cantidad de entradas no nulas en  $M$ .

De esta forma tenemos

$$u = 2 \cdot cv + r + n - F_0 \quad \text{y} \quad M_0 = n^2 - u.$$

Tenemos que cada una de las cantidades listadas anteriormente ( $cv, r, F_0$  y  $u$ ) dependen linealmente de  $n$ , más aún, se tiene que  $cv, r, F_0 < n$ . También, por el hecho de que cualquier matriz  $M$  es distinta de la matriz nula (es decir, siempre  $u > 0$ ) obtenemos  $M_0 < n^2$ .

## 2. Problema SING sobre $\mathbb{Z}$

En esta sección probaremos que el problema de decisión SING es indecidible para  $S = \mathbb{Z}$  y  $E = \{0, 1\}$ .

**Teorema 4.2: (Indecibilidad de SING sobre  $\mathbb{Z}$ )** Dada una matriz  $M = M(x_1, \dots, x_t)$  con entradas en  $\{0, 1\} \cup \{x_1, \dots, x_t\}$ , es indecible saber si existen  $a_1, \dots, a_t \in \mathbb{Z}$  tales que  $\det(M(a_1, \dots, a_t)) = 0$ .

**Demostración:** Para esta demostración buscaremos reducir el décimo problema de Hilbert al problema SING.

Elegimos una instancia del problema de Hilbert, es decir, una ecuación diofántica de la forma  $p(x_1, \dots, x_t) = 0$ . Construimos una fórmula para  $p$  usando solo  $+$ ,  $-$ ,  $\cdot$ ,  $1$ ,  $0$  y además reemplazamos cada constante  $c \geq 2$  con representación binaria  $c = \sum_{i=0}^l b_i 2^i$  por la fórmula:

$$(4.3) \quad b_0 + (1 + 1)[b_1 + (1 + 1)[b_2 + (1 + 1)[b_3 + \dots + (1 + 1)[b_l] \dots ]]].$$

Por la Proposición 4.1, la fórmula resultante  $f_p$  del polinomio  $p$ , la podemos convertir en una matriz  $M(x_1, \dots, x_t)$  tal que  $\det(M(x_1, \dots, x_t)) = p(x_1, \dots, x_t)$ . Con esto tenemos que si se pudiese resolver SING mediante algún algoritmo, entonces se podría resolver el problema de Hilbert, y como esto no es posible, entonces SING es indecible.

**Q. E. D.**

**Corolario 4.3:** El problema de decisión MINRANK es indecible sobre  $\mathbb{Z}$ .

**Demostración:** Al final de la Sección 3, Capítulo 2, mostramos que existe una reducción desde el problema de decisión SING al problema de decisión MINRANK. Además, usando el Teorema 4.2 tenemos que el problema de decisión MINRANK es indecible sobre  $\mathbb{Z}$  (ya que SING lo es).

**Q. E. D.**

## Teoría Existencial y Reducciones

### 1. Teoría existencial de primer orden

En esta sección describiremos la sintaxis de la teoría de primer orden existencial sobre cuerpos y mostraremos algunos resultados de complejidad para nuestros problemas de decisión. Más adelante en este trabajo aplicaremos esto a nuestros problemas de rango.

Sobre cualquier cuerpo  $\mathbb{F}$  tenemos los operadores aritméticos  $+, \cdot, -$ ; constantes  $0, 1$  y la relación de igualdad  $=$ . Agregando los operadores booleanos  $\wedge, \vee, \neg$  y el cuantificador existencial  $\exists$ , obtenemos el lenguaje de primer orden.

Definiremos los componentes de este vocabulario de la siguiente forma.

- Variables(V) :=  $x_1, x_2, \dots$
- Constantes(C) :=  $0, 1$ .
- Términos básicos(TB) := (V), (C).
- Término(T) := (TB), (T + T), (T · T).
- Fórmula de igualdad(FG) :=  $T = T$ .
- Fórmula Booleana(FB) := (FG), ( $\neg$ FB), (FB  $\wedge$  FB), (FB  $\vee$  FB).
- Fórmula(F) := (FB), ( $\exists V : F$ ).

Es importante acotar que estamos trabajando con el fragmento existencial de la lógica de primer orden de la teoría aritmética, es decir, todas nuestras fórmulas se encuentran en forma normal prenexa donde el único cuantificador admisible es el existencial.

Definiremos una *sentencia* como una fórmula que no posee variables libres (todas las variables están acotadas por el cuantificador existencial).

Decimos que una sentencia  $\phi$  es *verdadera* en el cuerpo  $\mathbb{F}$  (es decir, el cuerpo  $\mathbb{F}$  es un modelo de la sentencia  $\phi$ ) si la sentencia es verdadera cuando las variables cuantificadas son interpretados sobre elementos de  $\mathbb{F}$ , y a las operaciones aritméticas y a las constantes se les da su interpretación natural. Si  $\mathbb{F}$  es modelo de  $\phi$  lo representamos como  $\mathbb{F} \models \phi$ , de lo contrario,  $\mathbb{F} \not\models \phi$ .



A continuación mostraremos unos ejemplos.

$$\mathbb{Q} \models \exists x \exists y : x + y = -2 \wedge xy = -8$$

$$\mathbb{R} \not\models \exists x : x^2 + 1 = 0$$

Para un cuerpo  $\mathbb{F}$ , definimos la teoría existencial sobre  $\mathbb{F}$  como:

$$Eth(\mathbb{F}) = \{\phi : \mathbb{F} \models \phi\}$$

El problema de decisión  $Eth(\mathbb{F})$  es: dada la entrada  $\phi$  (una sentencia), averiguar si  $\mathbb{F} \models \phi$ .

**Proposición 5.1:** Para un cuerpo cualquiera  $\mathbb{F}$ ,  $Eth(\mathbb{F})$  es NP-fuerte.

**Demostración:** Reduciremos el problema desde 3-SAT. Sea  $C$  una instancia de 3-SAT, es decir,

$$C \equiv C_1 \wedge C_2 \wedge \dots \wedge C_k,$$

donde  $C_i \equiv (l_{i1} \vee l_{i2} \vee l_{i3})$  y  $l_{ij} \in \{y_1, y_2, \dots, y_t\} \cup \{\bar{y}_1, \bar{y}_2, \dots, \bar{y}_t\}$ ; donde  $y_i$  es un literal y  $\bar{y}_i$  es su negación, para cada  $i$ . Modificamos  $C$  para que se convierta en la fórmula aritmética  $f_C$  reemplazando cada  $y_i$  por la fórmula atómica  $x_i = 1$  y reemplazando cada  $\bar{y}_i$  por la fórmula atómica  $x_i = 0$ . De donde se obtiene que

$$C \text{ es satisfacible, si y solo si, } \mathbb{F} \models \exists x_1 \exists x_2 \dots \exists x_t : f_C.$$

El ser NP-fuerte se obtiene del hecho de que 3-SAT es NP-fuerte.

**Q. E. D.**

La complejidad de decidir  $Eth(\mathbb{F})$  depende del cuerpo  $\mathbb{F}$ . Existen unas cotas superiores de las que se tiene conocimiento.  $Eth(GF(q))$  está en NP, ya que se pueden sustituir las variables eligiendo de manera no determinista elementos del cuerpo y evaluando en la fórmula libre de variables resultante en tiempo polinomial. Por ahora es un problema abierto si  $Eth(\mathbb{Q})$  es decidible o no, en [9] se realiza con mayor profundidad un estudio sobre este problema.

## 2. Problemas de decisión sobre un cuerpo finito

En esta sección probaremos que tanto el problema SING como el problema NONSING sobre un cuerpo finito son tan difíciles de decidir como la correspondiente teoría existencial de primer orden. En particular, los cuatro problemas de decisión definidos son NP-completos.

**Lema 5.2:** Sea  $\mathbb{F} = GF(q)$  un cuerpo finito. Dada una sentencia existencial  $\exists x_1 \cdots \exists x_t : \phi(x_1, \dots, x_t)$  de longitud  $m$ , podemos construir dos matrices  $M'$  y  $M''$  de tamaño  $n \times n$  con entradas en  $\{0, 1\} \cup \{x_1, \dots, x_t\}$  en tiempo  $n^{\mathcal{O}(1)}$ , donde  $n = \mathcal{O}(mq)$ , tales que

$$\exists x_1 \cdots \exists x_t : \phi(x_1, \dots, x_t) \text{ sii } \exists (a_1, \dots, a_t) \in F^t : \det(M'(a_1, \dots, a_t)) = 0$$

y

$$\exists x_1 \cdots \exists x_t : \phi(x_1, \dots, x_t) \text{ sii } \exists (a_1, \dots, a_t) \in F^t : \det(M''(a_1, \dots, a_t)) \neq 0.$$

**Demostración:** Para construir la matriz  $M'$  modificamos el núcleo de la sentencia dada, es decir, la fórmula sin cuantificadores  $\phi$ , usando las reglas de re-escritura de la siguiente tabla:

$\mathbb{F} = GF(q)$	Reglas de re-escritura
Paso 1	$t(\mathbf{x}) = 0 \rightarrow (t(\mathbf{x}))^{q-1} = 0$
Paso 2	$\neg t(\mathbf{x}) = 0 \rightarrow 1 - t(\mathbf{x}) = 0$ $(t_1(\mathbf{x}) = 0) \vee (t_2(\mathbf{x}) = 0) \rightarrow t_1(\mathbf{x}) \cdot t_2(\mathbf{x}) = 0$ $(t_1(\mathbf{x}) = 0) \wedge (t_2(\mathbf{x}) = 0) \rightarrow 1 - (1 - t_1(\mathbf{x})) \cdot (1 - t_2(\mathbf{x})) = 0$
Paso 3	$t(\mathbf{x}) = 0 \rightarrow \det(M'(\mathbf{x})) = 0$

TABLA 4. Transformar una sentencia existencial a un problema de singularidad.

Inicialmente asumimos que cada fórmula lógica atómica es de la forma  $t(\mathbf{x}) = 0$ , para algún término aritmético  $t(\mathbf{x})$ . En el paso 1, usamos el hecho de que sobre el cuerpo  $GF(q)$ , la función  $x \mapsto x^{q-1}$  asigna el 0 al 0 y cualquier número no nulo al 1.

Después de este paso, podemos asumir que cualquier término aritmético toma solo los valores 0 y 1 sobre todas las posibles asignaciones de las variables. Esto nos facilita entender la validez de las tres reglas en el paso 2, ya que nos transforma operadores booleanos a operadores aritméticos, manteniendo el valor de verdad de la respectiva fórmula lógica. Después de que no se pueda aplicar más el paso 2, hemos comprimido  $\phi(\mathbf{x})$  a la fórmula atómica equivalente  $t(\mathbf{x}) = 0$ . En el paso 3, construimos la matriz  $M'$  tal que  $\det(M'(\mathbf{x})) = t(\mathbf{x})$  usando la Proposición 4.1.

Cuando usamos la reglas de re-escritura, cualquier término aritmético en el lado derecho de la regla, es una fórmula aritmética y debe permanecer como fórmula; es decir, no debe ser expandida como una suma de monomios, ya que dicha suma puede ser exponencialmente larga.

Del hecho de que la longitud de  $\phi$  es  $m$  y considerando el paso 1 de la Tabla 4, tenemos que cada fórmula atómica  $t(\mathbf{x})$  se transforma en  $(t(\mathbf{x}))^{q-1} = t(\mathbf{x}) \cdots t(\mathbf{x})$ ; de donde la longitud de  $\phi$  después de aplicar todos los pasos es lineal respecto a  $mq$ .

La construcción de la matriz  $M''$  es análoga, solo que se usan las reglas de re-escritura de la siguiente tabla:

$\mathbb{F} = GF(q)$	Reglas de re-escritura
Paso 1	$t(\mathbf{x}) = 0 \rightarrow 1 - (t(\mathbf{x}))^{q-1} \neq 0$
Paso 2	$\neg t(\mathbf{x}) \neq 0 \rightarrow 1 - t(\mathbf{x}) \neq 0$ $(t_1(\mathbf{x}) \neq 0) \vee (t_2(\mathbf{x}) \neq 0) \rightarrow 1 - (1 - t_1(\mathbf{x})) \cdot (1 - t_2(\mathbf{x})) \neq 0$ $(t_1(\mathbf{x}) \neq 0) \wedge (t_2(\mathbf{x}) \neq 0) \rightarrow t_1(\mathbf{x}) \cdot t_2(\mathbf{x}) \neq 0$
Paso 3	$t(\mathbf{x}) \neq 0 \rightarrow \det(M''(\mathbf{x})) \neq 0$

TABLA 5. Transformar una sentencia existencial a un problema de no singularidad.

### Q. E. D.

Mostraremos un ejemplo del proceso del lema anterior. Para esto tomaremos

$$\mathbb{F} = GF(3) \text{ y } \phi(x, y, z) = [(x - 2 = 0) \wedge (yz - 1 = 0)] \vee [\neg(z - 1 = 0)].$$

Aplicaremos las reglas de re-escritura en el siguiente orden: paso 1 (sustituir  $t(\mathbf{x}) = 0$  por  $(t(\mathbf{x}))^{q-1} = 0$ ), paso 2 (remover negaciones, remover conjunciones y remover disyunciones; en este orden). El paso 3 se omitirá ya que es el mismo de la Proposición 4.1. También es importante mencionar el hecho de que aplicamos las reglas de la Tabla 4 desde la operación más interna a la más externa.

$$\begin{aligned}
& [(x - 2 = 0) \wedge (yz - 1 = 0)] \vee [\neg(z - 1 = 0)] \\
& \quad \Downarrow \\
& \{[(x - 2)(x - 2) = 0] \wedge [(yz - 1)(yz - 1) = 0]\} \vee \{\neg[(z - 1)(z - 1) = 0]\} \\
& \quad \Downarrow \\
& \{[(x - 2)(x - 2) = 0] \wedge [(yz - 1)(yz - 1) = 0]\} \vee \{1 - (z - 1)(z - 1) = 0\} \\
& \quad \Downarrow \\
& \{1 - [1 - (x - 2)(x - 2)][1 - (yz - 1)(yz - 1)] = 0\} \vee \{1 - (z - 1)(z - 1) = 0\} \\
& \quad \Downarrow \\
& \{1 - [1 - (x - 2)(x - 2)][1 - (yz - 1)(yz - 1)]\} \{1 - (z - 1)(z - 1)\} = 0
\end{aligned}$$

A la ecuación resultante en el último paso es a la que aplicaremos la Proposición 4.1 para obtener la matriz deseada.

**Corolario 5.3:** Sea  $\mathbb{F}$  un cuerpo finito  $GF(q)$ . Para  $S = \mathbb{F}$  y  $\{0, 1\} \subseteq E \subseteq GF(q)$ , los problemas de decisión MAXRANK, NONSING, MINRANK y SING son NP-completos.

**Demostración:** Todos estos problemas están en NP, ya que de manera no determinista puede adivinarse una asignación de las variables tal que satisfaga el respectivo problema. Además, calcular el rango de una matriz sin variables (es decir, constante) se realiza en tiempo polinomial.

El ser NP-fuerte se obtiene de usar el Lema 5.2, que nos dice que resolver el problema de la teoría existencial es tan difícil como resolver los problemas SING y NONSING. Luego, la Proposición 5.1 nos dice que la teoría existencial es NP-fuerte; por lo cual, los problemas SING y NONSING son NP-fuertes. Con lo cual tenemos que son NP-completos.

Ahora, reduciremos el problema SING a MINRANK. Dada  $M$  una matriz  $n \times n$ , elegimos  $k = n - 1$ .

Si  $M$  es singular, para algún  $(a_1, \dots, a_t) \in \mathbb{F}^t$  el determinante es nulo, lo que implica que por lo menos una de las filas de la matriz no es linealmente independiente con el resto, por lo tanto el rango de  $M$  para algún  $(a_1, \dots, a_t) \in \mathbb{F}^t$  es a lo sumo  $n - 1$ . Así, con el valor de  $k$  seleccionado anteriormente, la respuesta de  $\text{minrank}_F(M) \leq k$  es sí. Si es no singular, entonces para todo  $(a_1, \dots, a_t) \in \mathbb{F}^t$  el determinante es no nulo, lo que implica que todas las filas de la matriz son linealmente independientes. Así, el rango de la matriz es  $n$  para todo  $(a_1, \dots, a_t) \in \mathbb{F}^t$ , de donde la respuesta a  $\text{minrank}_F(M) \leq k$  es no.

De manera análoga se reduce el problema NONSING a MAXRANK.

Con estas reducciones obtenemos que resolver MAXRANK y MINRANK implica resolver NONSING y SING respectivamente; pero estos últimos son NP-completos. Así, los cuatro problemas son NP-completos.

**Q. E. D.**

### 3. Cota inferior para SING sobre $\mathbb{Q}$ y $\mathbb{R}$

En esta sección, probaremos que el problema de decisión SING sobre cualquiera de los cuerpos  $\mathbb{Q}$  o  $\mathbb{R}$  es tan difícil de decidir como la correspondiente teoría existencial de primer orden. En particular, son NP-fuertes.

**Lema 5.4:** Sea  $\mathbb{F} = \mathbb{Q}$  o  $\mathbb{F} = \mathbb{R}$ . Dada una sentencia existencial  $\exists x_1 \cdots \exists x_t : \phi(x_1, \dots, x_t)$  de longitud  $m$ . Para  $n = \mathcal{O}(m)$  podemos construir en tiempo  $n^{\mathcal{O}(1)}$  una matriz  $M$  de tamaño  $n \times n$  con entradas en  $\{0, 1\} \cup \{x_1, \dots, x_t\}$  tal que

$$\exists x_1 \cdots \exists x_t : \phi(x_1, \dots, x_t) \text{ sii } \exists (a_1, \dots, a_t) \in \mathbb{F}^t : \det(M(a_1, \dots, a_t)) = 0.$$

**Demostración:** La demostración es análoga a la del Lema 5.2, solo que trabajaremos la negación de manera diferente.

Para construir la matriz  $M$ , modificaremos la fórmula no cuantificada  $\phi$  con las reglas de re-escritura de la Tabla 6.

$\mathbb{F} = \mathbb{Q}$ o $\mathbb{R}$	Reglas de re-escritura
Paso 1	$\neg(F_1 \wedge F_2) \rightarrow (\neg F_1) \vee (\neg F_2)$ $\neg(F_1 \vee F_2) \rightarrow (\neg F_1) \wedge (\neg F_2)$
Paso 2	$\neg t(\mathbf{x}) = 0 \rightarrow 1 - z \cdot t(\mathbf{x}) = 0$
Paso 3	$t(\mathbf{x}') = 0 \rightarrow (t(\mathbf{x}'))^2 = 0$
Paso 4	$(t_1(\mathbf{x}') = 0) \vee (t_2(\mathbf{x}') = 0) \rightarrow t_1(\mathbf{x}') \cdot t_2(\mathbf{x}') = 0$ $(t_1(\mathbf{x}') = 0) \wedge (t_2(\mathbf{x}') = 0) \rightarrow t_1(\mathbf{x}') + t_2(\mathbf{x}') = 0$
Paso 5	$t(\mathbf{x}') = 0 \rightarrow \det(M(\mathbf{x}')) = 0$

TABLA 6. Transformar una sentencia existencial a un problema de singularidad, sobre  $\mathbb{Q}$  y  $\mathbb{R}$ .

Los pasos 3, 4 y 5 de la Tabla 6 se corresponden con los pasos 1, 2 y 3 de la Tabla 4, excepto que la regla para la negación de la Tabla 4 no podemos usarla para esta demostración, ya que las variables pueden tomar muchos más valores que solamente 0 y 1. Los primeros dos pasos de la Tabla 6 sirven para remover la negación. En el paso 1, usamos las leyes de De Morgan para mover las negaciones para que estén junto a las fórmulas atómicas. En el paso 2, reemplazamos cada negación de una fórmula atómica por una fórmula no negada. Introducimos una nueva variable  $z$  por cada fórmula atómica como las antes mencionadas, que representa el inverso del término  $t(\mathbf{x})$ . Estas nuevas variables deben estar existencialmente cuantificadas. En el paso 3, usaremos el hecho de que tanto en  $\mathbb{Q}$  como en  $\mathbb{R}$  la función  $x \mapsto x^2$  es no negativa. Después del paso 3, podemos asumir que todo término aritmético toma solo valores no negativos bajo cualquier asignación posible de las variables. Esto último

nos permite asegurar la validez de las dos reglas en el paso 4 (de manera similar al Lema 5.2). Cuando no podamos aplicar más reglas de re-escritura del paso 4, entonces hemos comprimido a  $\phi(\mathbf{x})$  en una fórmula atómica equivalente  $t(\mathbf{x}') = 0$ . En el paso 5, construimos la matriz  $M$  tal que  $\det(M(\mathbf{x}')) = t(\mathbf{x}')$ , usando la Proposición 4.1.

**Q. E. D.**

**Corolario 5.5:** Sea  $\mathbb{F} = \mathbb{Q}$  o  $\mathbb{F} = \mathbb{R}$ . El problema de decisión SING para  $S = \mathbb{F}$  y  $E = \{0, 1\}$  es NP-fuerte.

**Demostración:** La demostración es análoga al corolario 5.3, solo que para este caso usamos el Lema 5.4 para reducir nuestro problema a *ETH* y luego la Proposición 5.1 para obtener el ser NP-fuerte.

**Q. E. D.**

#### 4. Cota inferior para MINRANK sobre un cuerpo

Hemos probado para los cuerpos  $GF(q)$ ,  $\mathbb{Q}$  y  $\mathbb{R}$  que el problema de decisión SING es tan difícil de decidir como la correspondiente teoría existencial de primer orden. Parece improbable que podamos generalizar este resultado para cualquier cuerpo, dado que se ha encontrado un algoritmo aleatorio en tiempo polinomial para SING sobre  $\mathbb{C}$  (Teorema 3.6) y la teoría existencial de primer orden es NP-fuerte sobre cualquier cuerpo, en particular sobre  $\mathbb{C}$ .

No obstante, solo un paso en las pruebas de los Lemas 5.2 y 5.4 hacen que no se pueda generalizar para cualquier cuerpo: la reducción de un sistema (conjunción) de ecuaciones a una única ecuación, lo cual es necesario para poder codificar una sentencia existencial general como un problema de singularidad. Sin embargo, podemos observar que un sistema de ecuaciones puede codificarse como un único problema MINRANK.

En esta sección mostraremos que sobre cualquier cuerpo el problema MINRANK es en efecto tan difícil de resolver como la correspondiente teoría existencial de primer orden. Nuestra construcción también nos llevará a una prueba alternativa para mostrar la dificultad del problema SING sobre los cuerpos  $GF(q)$ ,  $\mathbb{Q}$  y  $\mathbb{R}$ .

**Lema 5.6:** Sea  $\mathbb{F}$  un cuerpo. Dada una sentencia existencial  $\exists x_1 \cdots \exists x_t : \phi(x_1, \dots, x_t)$  de longitud  $m$ , podemos construir en tiempo  $m^{\mathcal{O}(1)}$  una sentencia existencial equivalente

$\exists x_1 \cdots \exists x_{t'} : \psi(x_1, \dots, x_{t'})$  tal que  $\psi$  no contiene negaciones ni disyunciones, es decir,  $\psi$  es una conjunción de fórmulas atómicas,

$$\psi(\mathbf{x}') \equiv (p_1(\mathbf{x}') = 0) \wedge \cdots \wedge (p_r(\mathbf{x}') = 0),$$

donde  $p_i$  es una fórmula aritmética, con  $i = 1, \dots, r$  y además

$$\mathbb{F} \models \exists \mathbf{x} : \phi(\mathbf{x}) \quad \text{sii} \quad \mathbb{F} \models \exists \mathbf{x}' : \psi(\mathbf{x}').$$

**Demostración:** Primero removemos todas las negaciones de  $\phi$  usando las reglas de re-escritura en los pasos 1 y 2 de la Tabla 6, que son válidos en cualquier cuerpo, obteniendo así  $\phi'$ .

Sin pérdida de generalidad, podemos asumir que la sentencia existencial dada es de la forma

$$\exists x_1 \cdots \exists x_t \exists y_1 \cdots \exists y_{t'} \cdot \phi'(x_1, \dots, x_t, y_1, \dots, y_{t'}),$$

donde  $\phi'$  es una fórmula sin cuantificadores y sin negaciones sobre las variables  $x_1, \dots, x_t, y_1, \dots, y_{t'}$ .

Digamos que  $\phi'$  tiene  $s$  sub-fórmulas  $f_1, \dots, f_s$ ; donde cada una de ellas puede ser atómica o compuesta. Para cada sub-fórmula  $f_i$ , introducimos una nueva variable  $z_i$  (existencialmente cuantificada) y construimos una nueva fórmula  $f'_i$  que bien es atómica o es la conjunción de dos fórmulas atómicas. Las  $f'_i$  son construidas de modo que

$$(5.1) \quad \begin{aligned} & \exists x_1 \cdots \exists x_t : \text{“}f_i \text{ es satisfacible”} \\ & \quad \quad \quad \updownarrow \\ & \exists x_1 \cdots \exists x_t \exists z_1 \cdots \exists z_s : \text{“}z_i = 0 \text{ y } f'_j \text{ es satisfacible para todas} \\ & \quad \quad \quad \text{las sub-fórmulas } f_j \text{ de } f_i \text{ (incluyendo } f_i \text{)”}. \end{aligned}$$

Si  $f_1$  es la sub-fórmula correspondiente a la fórmula entera  $\phi'$ , esto implica que

$$\begin{aligned} & \exists \mathbf{x} : \phi'(\mathbf{x}) \\ & \quad \quad \quad \updownarrow \\ & \exists \mathbf{x}, \mathbf{z} : (z_1 = 0) \wedge f'_1(\mathbf{x}, \mathbf{z}) \wedge \cdots \wedge f'_s(\mathbf{x}, \mathbf{z}). \end{aligned}$$

Para cada sub-fórmula original  $f_i$  la nueva fórmula  $f'_i$  es construida usando la siguiente tabla:

$f_i$	$f'_i$
$p_i(\mathbf{x}) = 0$	$p_i(\mathbf{x}) = z_i$
$f_j \vee f_k$	$z_j \cdot z_k = z_i$
$f_j \wedge f_k$	$(z_j \cdot z_k = z_i) \wedge (z_j + z_k = z_i)$

TABLA 7. Sub-construcción para la eliminación de  $\vee$ .

Haciendo inducción en la estructura de  $f_i$ , se puede comprobar que esta construcción sí satisface (5.1). El caso base, cuando  $f_i$  es atómica, se cumple naturalmente.

Supongamos que  $f_i = f_j \vee f_k$ , y que para  $f_j$  y  $f_k$  se cumple (5.1). Debemos probar la doble implicación de (5.1) para  $f_i$ .

( $\Leftarrow$ ) Supongamos que para algún cuerpo  $\mathbb{F}$  es cierto que

$$(5.2) \quad \exists x_1 \cdots \exists x_t \exists z_1 \cdots \exists z_s : "z_i = 0 \text{ y } f'_l \text{ es satisfacible para todas las sub-fórmulas } f_l \text{ de } f_i \text{ (incluyendo } f_i)".$$

Como  $z_i = 0$  y  $z_j \cdot z_k = z_i$  entonces  $z_j = 0$  o  $z_k = 0$ . Supongamos sin pérdida de generalidad que  $z_j = 0$ . Todas las sub-fórmulas de  $f_j$  son sub-fórmulas de  $f_i$  y por lo tanto se cumple que

$$(5.3) \quad \exists x_1 \cdots \exists x_t \exists z_1 \cdots \exists z_s : "z_j = 0 \text{ y } f'_l \text{ es satisfacible para todas las sub-fórmulas } f_l \text{ de } f_j \text{ (incluyendo } f_j)".$$

Por hipótesis inductiva, se tiene que  $\mathbb{F} \models \exists \mathbf{x} : f_j(\mathbf{x})$ , de donde obtenemos  $\mathbb{F} \models \exists \mathbf{x} : f_i(\mathbf{x})$ .

( $\Rightarrow$ ) Supongamos ahora que  $\mathbb{F} \models \exists \mathbf{x} : f_i(\mathbf{x})$ . Entonces  $\mathbb{F} \models \exists \mathbf{x} : f_j(\mathbf{x})$  o  $\mathbb{F} \models \exists \mathbf{x} : f_k(\mathbf{x})$ .

Supongamos que ocurre lo primero. Por hipótesis inductiva, es cierto (5.3) y por lo tanto podemos deducir  $z_i = 0$ . Además, todas las fórmulas  $f'_l$  se satisfacen, con  $f_l$  sub-fórmula de  $f_i$ . Hay tres casos,  $f_l$  es sub-fórmula de  $f_j$ , o de  $f_k$ , o es igual a  $f_j \vee f_k$ . El primer caso está listo, por ser parte de la hipótesis inductiva, el tercer caso también, porque es justamente la fórmula  $z_j \cdot z_k = z_i$ , solo falta ver que para el segundo caso también es cierto. Note que las variables  $z_l$  simplemente se pueden definir en función de la escogencia de las variables  $\mathbf{x}$ , es decir, si  $(a_1, \dots, a_t) \in \mathbb{F}$ , es una interpretación de las variables  $\mathbf{x} = (x_1, \dots, x_t)$ , definimos el valor de  $z_l$  inductivamente siguiendo la reglas de la Tabla 7, para que así se satisfagan cada una de las sub-fórmulas de  $f_k$ .

Por último, supongamos que  $f_i = f_j \wedge f_k$ , y que para  $f_j$  y  $f_k$  se cumple (5.1). Probaremos la doble implicación de (5.1) para  $f_i$ .



( $\Leftarrow$ ) Supongamos que para algún cuerpo  $\mathbb{F}$  se cumple (5.2). Como  $z_i = 0$  y  $z_i = z_j \cdot z_k$  entonces  $z_i = 0$  o  $z_k = 0$ , supongamos (sin pérdida de generalidad) que  $z_j = 0$ . Además tenemos que  $z_i = z_j + z_k$ , por lo cual  $z_k = 0$ . Como toda sub-fórmula de  $f_j$  y de  $f_k$  son sub-fórmulas de  $f_i$ , se cumple (5.3) para  $f_j$  y  $f_k$ . Luego por hipótesis inductiva se tiene que  $\mathbb{F} \models \exists \mathbf{x} : f_j(\mathbf{x})$  y  $\mathbb{F} \models \exists \mathbf{x} : f_k(\mathbf{x})$ . De donde,  $\mathbb{F} \models \exists \mathbf{x} : f_i(\mathbf{x})$ .

( $\Rightarrow$ ) Supongamos ahora que  $\mathbb{F} \models \exists \mathbf{x} : f_i(\mathbf{x})$ . Entonces  $\mathbb{F} \models \exists \mathbf{x} : f_j(\mathbf{x})$  y  $\mathbb{F} \models \exists \mathbf{x} : f_k(\mathbf{x})$ . Por hipótesis inductiva, es cierto (5.3) para  $f_j$  y  $f_k$ , de esto deducimos  $z_i = 0$ . Además, todas las fórmulas  $f'_i$  se satisfacen, con  $f_i$  sub-fórmula de  $f_i$ . Hay tres casos,  $f_i$  es sub-fórmula de  $f_j$ , o de  $f_k$ , o es  $f_j \wedge f_k$ . El primer y segundo caso son inmediatos por la hipótesis inductiva, el tercer caso también por ser el caso  $(z_i = z_j \cdot z_k) \wedge (z_i = z_j + z_k)$ . Así, se satisface (5.2).

**Q. E. D.**

**Lema 5.7:** Sea  $\mathbb{F}$  un cuerpo y  $\phi$  una sentencia existencial de longitud  $m$ . Para  $n = \mathcal{O}(m)$  podemos construir en tiempo  $n^{\mathcal{O}(1)}$  un entero  $k$  y una matriz  $n \times n$  con entradas en  $\{0, 1\} \cup \{x_1, \dots, x_t\}$ , tal que

$$\text{minrank}_{\mathbb{F}}(M) \leq k \quad \text{sii} \quad \mathbb{F} \models \phi.$$

**Demostración:** Sea una sentencia existencial dada. Primero removemos todas las negaciones y disjunciones usando la construcción del Lema 5.6.

Sin pérdida de generalidad, podemos asumir que la sentencia existencial dada es

$$(5.4) \quad \exists \mathbf{x} : (p_1(\mathbf{x}) = 0) \wedge \dots \wedge (p_r(\mathbf{x}) = 0),$$

para ciertas fórmulas aritméticas  $p_i$ , con  $i = 1, \dots, r$ . Por la Proposición 4.1, para cada  $p_i(x_1, \dots, x_t)$  podemos encontrar una matriz  $M_i$  de tamaño  $n_i \times n_i$  con entradas en  $\{0, 1\} \cup \{x_1, \dots, x_t\}$  tal que  $\det(M_i(x_1, \dots, x_t)) = p_i(x_1, \dots, x_t)$  y  $\text{minrank}_{\mathbb{F}}(M_i) \geq n_i - 1$ .

Sean  $n = \sum_{i=1}^r n_i$  y  $k = \sum_{i=1}^r (n_i - 1)$ . Construimos la matriz  $M$  de tamaño  $n \times n$  tal que  $M = \text{diag}(M_1, \dots, M_r)$ .

Claramente,  $\text{minrank}_{\mathbb{F}}(M) \geq k$ . Además,  $\text{rank}(M) = k$  solo cuando los polinomios  $p_i$  son cero simultáneamente en algún  $(a_1, \dots, a_t) \in \mathbb{F}$ , lo que implica que se satisface (5.4). De esta forma, el rango de cada una de las matrices  $M_i$  será  $n_i - 1$  para  $(a_1, \dots, a_t)$ . Así,  $\text{minrank}_{\mathbb{F}}(M) \leq k$  si y solo si  $\mathbb{F} \models \phi$ .

**Q. E. D.**

**Corolario 5.8:** Sea  $\mathbb{F}$  un cuerpo. El problema de decisión MINRANK para  $S = \mathbb{F}$  y  $E = \{0, 1\}$  es NP-fuerte.

**Demostración:** El resultado se obtiene usando el Lema 5.7 y luego aplicando la Proposición 5.1.

**Q. E. D.**

El siguiente lema es una demostración alterna del Lema 5.2 y del Lema 5.4, además de incluir un resultado cuando  $\mathbb{F}$  es el cuerpo de los racionales o los reales.

**Lema 5.9:** Sea  $\mathbb{F} = GF(q)$ . Dada una sentencia existencial  $\exists x_1 \cdots \exists x_t : \phi(x_1, \dots, x_t)$  de longitud  $m$ , podemos en tiempo  $n^{\mathcal{O}(1)}$  construir dos matrices  $M'$  y  $M''$  con entradas en  $\{0, 1\} \cup \{x_1, \dots, x_t\}$  y de orden  $n \times n$ , con  $n = \mathcal{O}(mq)$ , tales que

$$\mathbb{F} \models \exists x_1 \cdots \exists x_t : \phi(x_1, \dots, x_t) \quad \text{sii} \quad \exists (a_1, \dots, a_t) \in \mathbb{F}^t : \det(M'(a_1, \dots, a_t)) = 0$$

y

$$\mathbb{F} \models \exists x_1 \cdots \exists x_t : \phi(x_1, \dots, x_t) \quad \text{sii} \quad \exists (a_1, \dots, a_t) \in \mathbb{F}^t : \det(M''(a_1, \dots, a_t)) \neq 0.$$

Si  $\mathbb{F} = \mathbb{Q}$  o  $\mathbb{F} = \mathbb{R}$ , podemos construir en tiempo  $n^{\mathcal{O}(1)}$ , con  $n = \mathcal{O}(m)$ , una matriz  $M$  con entradas en  $\{0, 1\} \cup \{x_1, \dots, x_t\}$  y de orden  $n \times n$  tal que

$$\mathbb{F} \models \exists x_1 \cdots \exists x_t : \phi(x_1, \dots, x_t) \quad \text{sii} \quad \exists (a_1, \dots, a_t) \in \mathbb{F}^t : \det(M(a_1, \dots, a_t)) = 0.$$

**Demostración:** Consideremos una sentencia existencial como la dada en el enunciado. Primero removemos todas las negaciones y disjunciones usando la construcción del Lema 5.6. Hecho esto podemos asumir, sin pérdida de generalidad, que la fórmula existencial dada es de la forma

$$\exists \mathbf{x} : (p_1(\mathbf{x}) = 0) \wedge \cdots \wedge (p_r(\mathbf{x}) = 0)$$

para alguna fórmula aritmética  $p_i$ ,  $i = 1, \dots, r$ .

Si  $\mathbb{F}$  es un cuerpo finito  $GF(q)$ , usamos la propiedad de la función  $x \mapsto x^{q-1}$  que asigna el 0 al 0 y cualquier número no nulo al 1; es decir,

$$\begin{aligned} GF(q) \models \exists \mathbf{x} : (p_1(\mathbf{x}) = 0) \wedge \cdots \wedge (p_r(\mathbf{x}) = 0) \\ \iff \\ GF(q) \models \exists \mathbf{x} : 1 - (1 - (p_1(\mathbf{x}))^{q-1}) \cdot \dots \cdot (1 - (p_r(\mathbf{x}))^{q-1}) = 0 \\ \iff \\ GF(q) \models \exists \mathbf{x} : (1 - (p_1(\mathbf{x}))^{q-1}) \cdot \dots \cdot (1 - (p_r(\mathbf{x}))^{q-1}) \neq 0. \end{aligned}$$

Si  $\mathbb{F}$  es uno de los cuerpos  $\mathbb{Q}$  o  $\mathbb{R}$ , usamos la propiedad de la función  $x \mapsto x^2$  que asigna el 0 al 0 y a cualquier número no nulo le asigna un número positivo; es decir,

$$\begin{aligned} \mathbb{F} \models \exists \mathbf{x} : (p_1(\mathbf{x}) = 0) \wedge \cdots \wedge (p_r(\mathbf{x}) = 0) \\ \Updownarrow \\ \mathbb{F} \models \exists \mathbf{x} : (p_1(\mathbf{x}))^2 + \dots + (p_r(\mathbf{x}))^2 = 0. \end{aligned}$$

Ahora, podemos construir las matrices con las propiedades mencionadas en el enunciado de este lema usando la Proposición 4.1.

**Q. E. D.**

## 5. Cotas superiores para MINRANK sobre un cuerpo

En esta sección probaremos que el problema de decisión MINRANK es tan difícil de decidir como la teoría existencial de primer orden correspondiente. Combinado con los resultados anteriores tendremos que el problema de decisión MINRANK es de hecho equivalente (bajo transformaciones en tiempo polinomial) a decidir la teoría existencial de primer orden correspondiente.

Primero daremos la reducción para las matrices que solo usan las constantes 0 y 1, luego extenderemos el resultado a constantes más generales.

**Lema 5.10:** Sea  $\mathbb{F}$  un cuerpo. Dada una matriz  $M$  de orden  $n \times n$  con entradas en  $\{0, 1\} \cup \{x_1, \dots, x_t\}$  y algún  $k \leq n$ , podemos construir en tiempo  $n^{\mathcal{O}(1)}$  una sentencia existencial  $\phi$  tal que

$$\text{minrank}_{\mathbb{F}}(M) \leq k \quad \text{sii} \quad \mathbb{F} \models \phi.$$

**Demostración:** Podemos expresar con una sentencia existencial de primer orden la afirmación de que  $k$  columnas de  $M$  pueden generar a todas las columnas de  $M$ . Para esto introduciremos  $n$  variables nuevas  $y_1, \dots, y_n$  en conjunto con las variables que ya aparecen en  $M$ .

Construimos la matriz modificada  $M'$ , donde  $M'_{ij} = y_j \cdot M_{ij}$ ; es decir, cada columna de  $M'$  es una múltiplo (posiblemente nulo) de la columna correspondiente en  $M$ . También introduciremos además  $n^2$  variables nuevas  $z_{11}, \dots, z_{nn}$ , que forman una matriz  $Z$  de tamaño  $n \times n$ .

La afirmación  $\text{minrank}(M) \leq k$  es ahora equivalente a la siguiente:

“Es posible elegir  $y_j$  y  $z_{ij}$  de tal forma que a lo sumo  $k$  de los  $y_j$  son no nulos y la ecuación matricial  $M' \cdot Z = M$  se cumple.”

Nuestra sentencia va a ser una cuantificación existencial de una conjunción de dos fórmulas. La primera,  $f_1$ , dirá que a lo sumo  $k$  de los  $y_j$  son no nulos. La segunda,  $f_2$ , afirmará que la ecuación matricial  $M' \cdot Z = M$  se cumple.

Construcción de  $f_1$ : Usamos las funciones simétricas elementales definidas por

$$(5.5) \quad \sigma_j(y_1, \dots, y_n) = \sum_{A \subseteq \{1, \dots, n\} \wedge |A|=j} \prod_{i \in A} y_i$$

para  $j = 1, \dots, n$ . Estas funciones satisfacen la siguiente propiedad: Hay a lo sumo  $k$  variables  $y_j$  que son no nulos, si y solo si,

$$(5.6) \quad (\sigma_{k+1}(y_1, \dots, y_n) = 0) \wedge (\sigma_{k+2}(y_1, \dots, y_n) = 0) \wedge \dots \wedge (\sigma_n(y_1, \dots, y_n) = 0).$$

La implicación ( $\Rightarrow$ ) es directa. Para demostrar ( $\Leftarrow$ ) usaremos inducción y para los fundamentos de la inducción consideramos  $\sigma_n(y_1, \dots, y_n) = \prod_{i=1}^n y_i$ .

Como  $\mathbb{F}$  es un cuerpo, tenemos que cualquier posible divisor de  $\sigma_n(y_1, \dots, y_n)$  es no nulo, y como  $\sigma_n(y_1, \dots, y_n) = 0$  tenemos que alguno de los  $y_i = 0$ . Sin pérdida de generalidad podemos asumir que  $y_n = 0$ . De donde  $\sigma_{n-1}(y_1, \dots, y_n) = \prod_{i=1}^{n-1} y_i$  y el argumento usado anteriormente puede ser usado otra vez para demostrar que  $y_{n-1} = 0$ . Aplicando esto para cada función  $\sigma$  en orden decreciente obtenemos que a lo sumo  $k$  de las variables  $y_i$  son no nulas.

Necesitamos encontrar una fórmula corta para poder expresar  $\sigma_j(y_1, \dots, y_n) = 0$ . Consideremos el polinomio  $p(z, y_1, \dots, y_n) = (z + y_1) \cdots (z + y_n)$ . Demostraremos por inducción sobre  $n$  que  $(z + y_1) \cdots (z + y_n) = z^n + \sigma_1(y_1, \dots, y_n)z^{n-1} + \dots + \sigma_n(y_1, \dots, y_n)$ .

Suponemos  $n = 1$ . Sea  $p(z, y_1) = z + y_1$ , por definición de  $\sigma$  tenemos que  $\sigma_1(y_1) = y_1$ ; así,  $p(z, y_1) = z + \sigma_1(y_1)$ . Ahora lo suponemos cierto para  $n < k$  y lo demostramos para  $n = k$ . Sea  $p(z, y_1, \dots, y_k) = (z + y_1) \cdots (z + y_k)$ , entonces

$$\begin{aligned} p(z, y_1, \dots, y_k) &= (z + y_1) \cdots (z + y_{k-1})(z + y_k) \\ &\stackrel{\text{H.I.}}{=} (z^{k-1} + \sigma_1(y_1, \dots, y_{k-1})z^{k-2} + \dots + \sigma_{k-1}(y_1, \dots, y_{k-1}))(z + y_k) \\ &= z^k + \sigma_1(y_1, \dots, y_{k-1})z^{k-1} + \dots + \sigma_{k-1}(y_1, \dots, y_{k-1})z + y_k z^{k-1} + \end{aligned}$$

$$\begin{aligned}
& + y_k \sigma_1(y_1, \dots, y_{k-1}) z^{k-2} + \dots + y_k \sigma_{k-1}(y_1, \dots, y_{k-1}) \\
& = z^k + [\sigma_1(y_1, \dots, y_{k-1}) + y_k] z^{k-1} + \dots + y_k \sigma_{k-1}(y_1, \dots, y_{k-1}) \\
& = z^k + \sigma_1(y_1, \dots, y_k) z^{k-1} + \dots + \sigma_k(y_1, \dots, y_k)
\end{aligned}$$

Para poder transformar este polinomio en sentencia (y por ende tener  $\sigma_j(y_1, \dots, y_n)$  como sentencia), iremos escribiendo distintas sentencias ( $h_k$ ) mientras revolsemos la distributiva de los distintos  $(z + y_i)$  (es decir, mientras obtenemos la versión extendida de  $p$ ). El proceso que realizaremos es el siguiente: Al resolver la distributiva, crearemos la fórmula atómica  $h_i \equiv w = u + v$  (respectivamente  $w = u \cdot v$ ) si en el  $i$ -ésimo paso resolvemos una suma (o una multiplicación, respectivamente). Sin pérdida de generalidad, podemos asumir que las variables usadas en este proceso son  $w_1, \dots, w_s$  en adición a las variables iniciales  $y_1, \dots, y_n$  y las variables resultantes  $S_1, \dots, S_n$  que simularían a los respectivos  $\sigma_1, \dots, \sigma_n$ . Así, tenemos que la fórmula  $f_1$  es

$$f_1 \equiv h_1 \wedge \dots \wedge h_s \wedge (s_{k+1} = 0) \wedge \dots \wedge (s_n = 0).$$

Construcción de  $f_2$ : Necesitamos expresar  $M' \cdot Z = M$ . La entrada  $ij$  en la matriz producto es  $\sum_{k=1}^n y_k M_{ik} Z_{kj}$ . Más aún, construir la fórmula

$$(5.7) \quad g_{ij} \equiv y_1 M_{i1} Z_{1j} + \dots + y_n M_{in} Z_{nj} = M_{ij},$$

y definir

$$f_2 \equiv g_{11} \wedge g_{12} \wedge \dots \wedge g_{nn}$$

La sentencia requerida por el lema es

$$\text{minrank}(M) \leq k \equiv$$

$$\exists x_1 \dots \exists x_t \exists y_1 \dots \exists y_n \exists z_{11} \dots \exists z_{nn} \exists w_1 \dots \exists w_s \exists s_1 \dots \exists s_n : f_1 \wedge f_2.$$

### Q. E. D.

Restringimos las constantes en nuestra sentencia existencial a 0 y 1. Un análogo del Lema 5.10 también se cumple para MINRANK donde las matrices contienen constantes algebraicas, ya que las constantes algebraicas pueden ser definidas por sentencias cortas de primer orden.

- Sobre un cuerpo, la constante 2 puede definirse por

$$\phi(x) \equiv x = 1 + 1$$

- Sobre un cuerpo con característica 0, podemos definir la constante  $-\frac{4}{3}$  por

$$\phi(x) \equiv x \cdot (1 + 1 + 1) + 1 + 1 + 1 + 1 = 0$$

Pero esta sentencia no es eficiente en su escritura, ya que de tener constantes grandes, por ejemplo 1.000.000, tendremos esa misma cantidad de 1. Por lo cual usaremos una re-escritura de la sentencia usando nuevas variables  $x, y, z, \dots$  para representar las potencias de 2 (2, 4, 8, ...). Así, nuestra nueva sentencia es:

$$\phi(x) \equiv \exists y \exists z : (x \cdot (1 + y) + z = 0) \wedge (y = 1 + 1) \wedge (z = y + y)$$

Nos referiremos a esta forma de escribir las sentencias como la técnica de la representación doble.

- Sobre  $\mathbb{R}$ , la constante  $\sqrt{3}$  se puede definir por

$$\phi(x) \equiv \exists y : (x \cdot x = 1 + 1 + 1) \wedge (y \cdot y = x)$$

La última parte nos asegura la raíz cuadrada positiva. Escrita en forma de representación doble.

$$\phi(x) \equiv \exists y \exists z : (x \cdot x = 1 + z) \wedge (y \cdot y = x) \wedge (z = 1 + 1)$$

- Sobre un cuerpo, la constante 15 se define por

$$\phi(x) \equiv \exists y \exists z \exists w : (x = 1 + y + z + w) \wedge (y = 1 + 1) \wedge (z = y + y) \wedge (w = z + z)$$

- Sobre  $\mathbb{C}$ , las constantes  $i$  y  $-i$  están definidas por

$$\phi(x) \equiv \exists y : (x \cdot x + 1 = 0) \wedge (y \cdot y + 1 = 0) \wedge (x + y = 0)$$

Nótese que  $i$  y  $-i$  no pueden ser definidas por separado, ya que  $i$  solo puede ser definido por conjugación.

Si  $\mathbb{F}$  es un cuerpo, definimos el *cuerpo primo* de  $\mathbb{F}$  como la intersección de todos los sub-cuerpos de  $\mathbb{F}$ . Claramente el cuerpo primo subyacente de  $\mathbb{C}$  y  $\mathbb{R}$  es  $\mathbb{Q}$ ; y  $GF(q)$  es una extensión algebraica de dimensión finita de su cuerpo primo subyacente (que es  $GF(p)$ , para algún primo  $p$ ).

Para un cuerpo  $\mathbb{F}$ , definimos  $A_{\mathbb{F}}$  como el conjunto de todos los números que son algebraicos sobre el cuerpo primo subyacente a  $\mathbb{F}$ .

**Proposición 5.11:** Sea  $\mathbb{P}$  un cuerpo primo y  $e \in A_{\mathbb{P}}$ . Sea  $\mathbb{F}$  el cuerpo extensión más pequeño que contiene la constante  $e$ , y además nos es dada una representación estándar de  $\mathbb{F}$  como un espacio vectorial de dimensión  $k$  sobre  $\mathbb{P}$  (con la aritmética vectorial definida usando polinomios irreducibles). También es dada la representación de la constante  $e$  como un vector de números binarios, es decir, su representación binaria.

Es posible construir una fórmula existencial de primer orden  $\phi(x)$  definiendo  $e$  en tiempo polinomial, en función de la longitud de los bits de la representación de  $e$ .

**Demostración:** Sea  $\mathbb{P}$  un cuerpo primo y sea  $e \in A_{\mathbb{P}}$ . Sea  $\mathbb{F}$  el cuerpo extensión de  $\mathbb{P}$  más pequeño que contiene a  $e$ , donde  $\mathbb{F}$  es un  $\mathbb{P}$ -espacio de dimensión  $k$ . Además, nos es dada la representación binaria de  $e$  en términos de la base de  $\mathbb{F}$ .

Sea  $v_1, \dots, v_k$  una base de  $\mathbb{F}$  como  $\mathbb{P}$ -espacio y  $p_1, \dots, p_k \in \mathbb{P}[x]$  los polinomios irreducibles asociados a estos elementos respectivamente. Sin pérdida de generalidad, podemos asumir que  $p_i$  tiene coeficientes enteros para todo  $i$ .

Como  $e$  pertenece a  $\mathbb{F}$ , tenemos que  $e = \lambda_1 v_1 + \dots + \lambda_k v_k$  para ciertos  $\lambda_1, \dots, \lambda_k \in \mathbb{P}$  y como conocemos la representación binaria de  $e$  entonces conocemos los polinomios de grado 1  $q_1, \dots, q_k \in \mathbb{P}[x]$  que generan a  $\lambda_1, \dots, \lambda_k$ . Podemos asumir sin pérdida de generalidad que  $q_j$  es un polinomio a coeficientes enteros para todo  $j$ . (Como en el caso del número  $-\frac{4}{3}$ , en los ejemplos anteriores.)

Para todo  $p_i$  y  $q_j$  consideramos polinomios en varias variables que se obtienen de substituir los coeficientes de  $p_i$  y  $q_j$  por la correspondiente representación doble. A tales polinomios los denotamos por  $P_i(\mathbf{y}_i)$  y  $Q_j(\mathbf{x}_j)$  donde  $\mathbf{x}_j = (x_j, z_1, \dots, z_r)$  y  $\mathbf{y}_i = (y_i, z_1, \dots, z_r)$  para  $1 \leq i, j \leq k$  y  $z_1, \dots, z_r$  son las variables creadas al aplicar la técnica de representación doble, es decir,  $\mathbf{x}_j$  y  $\mathbf{y}_i$  son tales que si substituímos las variables  $z_1, \dots, z_r$  por sus respectivos valores en  $\mathbb{P}$  obtenemos que  $P_i(\mathbf{y}_i) = p_i(y_i)$  y  $Q_j(\mathbf{x}_j) = q_j(x_j)$ .

Ahora podemos construir la fórmula

$$\begin{aligned} \phi(x_1, \dots, x_k) \equiv & \exists y_1 \cdots \exists y_k \exists z_1 \cdots \exists z_r : \\ & (Q_1(\mathbf{x}_1) = 0) \wedge (P_1(\mathbf{y}_1) = 0) \wedge \cdots \\ & \cdots \wedge (Q_k(\mathbf{x}_k) = 0) \wedge (P_k(\mathbf{y}_k) = 0) \wedge \\ & (z_1 = 1 + 1) \wedge (z_2 = z_1 + z_1) \wedge \cdots \wedge (z_r = z_{r-1} + z_{r-1}). \end{aligned}$$

En  $\phi$  se realiza la creación de variables con base en la representación binaria de  $e$ , donde por cada bit se crea una nueva variable y cada nueva variable está relacionada con la anterior (debido a la representación doble). Asumimos que todas las constantes están representadas en  $r$  bits. Este proceso depende de la representación binaria de  $e$ . Con esto obtenemos el resultado de que se puede construir  $\phi(x)$  en tiempo polinomial, en función de la longitud de los bits de la representación de  $e$ .

**Q. E. D.**

La generalización del Lema 5.10 es la siguiente:

**Lema 5.12:** Sea  $\mathbb{F}$  un cuerpo. Sea  $\mathbb{F}'$  una extensión algebraica de dimensión finita del cuerpo primo subyacente a  $\mathbb{F}$ . Sea  $E \subseteq \mathbb{F}' (\subseteq A_{\mathbb{F}})$ .

Dada  $M$  una matriz  $n \times n$  con entradas en  $E \cup \{x_1, \dots, x_t\}$ , y un  $k \leq n$ , podemos en tiempo  $(ns)^{\mathcal{O}(1)}$  construir una sentencia existencial  $\phi$  tal que

$$\text{minrank}_{\mathbb{F}}(M) \leq k \quad \text{sii} \quad \mathbb{F} \models \phi,$$

donde  $s$  denota la máxima longitud en bits de la representación de una entrada de  $M$  (usando números binarios/coeficientes para elementos del cuerpo primo y vectores de estos para los números algebraicos).

**Demostración:** Usando la construcción de la demostración del Lema 5.10 combinado con la construcción de la Proposición 5.11.

**Q. E. D.**

**Corolario 5.13:** Sea  $\mathbb{F}$  un cuerpo. Sea  $\mathbb{F}'$  una extensión algebraica de dimensión finita del cuerpo primo subyacente a  $\mathbb{F}$ . Sea  $S = \mathbb{F}$  y sea  $\{0, 1\} \subseteq E \subseteq \mathbb{F}'$ .

El problema de decisión MINRANK es equivalente (a través de reducciones de tiempo polinomial) a decidir  $ETH(\mathbb{F})$ .

Si  $\mathbb{F} = \mathbb{Q}$  o  $\mathbb{F} = \mathbb{R}$ , entonces el problema de decisión SING y MINRANK son equivalentes a través de reducciones de tiempo polinomial.

**Demostración:** Es inmediato usando los Lemas 5.12, 5.7 y 5.4.

**Q. E. D.**



## Ocurrencia de las variables una única vez

En secciones anteriores se han considerado matrices  $M(x_1, \dots, x_t)$  con entradas en  $E \cup \{x_1, \dots, x_t\}$  y cada variable puede ocurrir una cantidad cualquiera de veces en  $M$ . En esta sección restringiremos la cantidad de veces que puede aparecer cada variable en la matriz a solo una vez. Llamaremos a estas matrices  $cvuv$  (cada variable una vez). En esta parte emplearemos la Definición 1.4 (polinomio multi-afín). Nótese que el determinante de una matriz  $cvuv$  es un polinomio multi-afín.

El siguiente lema es un equivalente del Lema 3.4, solo que considerando al polinomio  $p$  como multi-afín y a  $\mathbb{F}$  como un cuerpo cualquiera (no necesariamente infinito).

**Lema 6.1:** Sea  $p$  un polinomio multi-afín sobre el cuerpo  $\mathbb{F}$ .  $p$  es idénticamente cero sobre  $\mathbb{F}$ , si y solo si,  $p$  es el polinomio cero.

**Demostración:** ( $\Leftarrow$ ) Si  $p$  es el polinomio cero es evidente que es idénticamente cero.

Para demostrar el otro sentido de la demostración ( $\Rightarrow$ ), lo haremos por contra-recíproco: si  $p$  no es el polinomio nulo entonces  $p$  no es idénticamente cero.

Sea  $p$  un polinomio no nulo. Aplicaremos inducción sobre  $t$ , el número de variables; para probar que  $p$  no es idénticamente cero.

Si  $t = 1$ , entonces  $p(x) = ax + b$ , y alguno entre  $a$  o  $b$  es no nulo. Si  $b$  es no nulo, basta con elegir  $x = 0$  para tener un resultado distinto de cero. Si  $a$  es no nulo, elegimos  $x = \frac{1-b}{a}$  para obtener el valor 1. Así,  $p(x)$  no es idénticamente cero.

Suponemos que el resultado es cierto para  $t < k$  y lo demostramos para  $t = k$  variables. Sea  $p(x_1, \dots, x_k) = q(x_2, \dots, x_k)x_1 + r(x_2, \dots, x_k)$ ; donde  $q, r$  son polinomios multi-afines de  $k - 1$  variables. Como  $p$  no es el polinomio cero, alguno entre  $q$  o  $r$  debe ser distinto del polinomio cero. Luego por hipótesis inductiva, existe  $(a_2, \dots, a_k)$  tal que alguno entre  $q$  o  $r$  debe tomar un valor no nulo. Usando dicha sustitución en  $q$  y  $r$  obtenemos  $ax_1 + b$ ; donde  $a$  y  $b$ , no pueden ser cero al mismo tiempo. Luego, como en el caso básico ( $t = 1$ ), tenemos que  $ax_1 + b$  debe tomar algún valor no nulo en  $\mathbb{F}$ .

**Q. E. D.**

Note que este lema no es necesariamente cierto para polinomios cuyas variables ocurran con un grado mayor a 1. Por ejemplo, para  $p(x) = x^2 - x$  no es el polinomio cero, pero sí es idénticamente cero sobre el cuerpo  $GF(2)$ .

**Corolario 6.2:** Un polinomio multi-afín es idénticamente cero sobre un cuerpo  $\mathbb{F}$ , si y solo si, es idénticamente cero sobre un cuerpo extensión  $\mathbb{F}' \supseteq \mathbb{F}$ .

**Demostración:** ( $\Leftarrow$ ) Si  $p$  es idénticamente cero sobre  $\mathbb{F}' \supseteq \mathbb{F}$ , es directo el hecho de que  $p$  es idénticamente cero sobre  $\mathbb{F}$ .

( $\Rightarrow$ ) Suponemos  $p$  idénticamente cero sobre  $\mathbb{F}$ . Por el Lema 6.1 tenemos que  $p$  es el polinomio cero. De donde,  $p$  es el polinomio cero sobre  $\mathbb{F}'$ . Por lo tanto,  $p$  es idénticamente cero sobre  $\mathbb{F}'$ .

**Q. E. D.**

**Lema 6.3:** Una función multi-afín sobre un cuerpo  $\mathbb{F}$  es constante o toma todos los valores de  $\mathbb{F}$ .

**Demostración:** Suponemos que la función es no constante y probaremos que toma todos los valores de  $\mathbb{F}$ .

Aplicaremos inducción sobre  $t$ , el número de variables. Suponemos  $t = 1$ . Tenemos  $p(x) = ax + b$ . Como  $p$  es no constante, entonces  $a \neq 0$ . Luego, para obtener  $p(x) = c$  elegimos  $x = \frac{c-b}{a}$ .

Ahora suponemos cierto el resultado para  $t < k$  y lo demostramos para el caso  $t = k$  variables. Elegimos cualquier variable, digamos  $x$ ; escribimos  $p(x, x_2, \dots, x_k) = a(x_2, \dots, x_k)x + b(x_2, \dots, x_k)$ ; donde  $a$  y  $b$  son polinomios multi-afines de  $k - 1$  variables. Los polinomios  $a$  y  $b$  no pueden ser los dos constantes. Si  $a$  es constante, elegimos una asignación de variables que haga que  $b$  tome el valor  $b'$ . De donde tenemos  $p(x) = ax + b'$ . Para obtener  $p(x) = c$ , elegimos  $x = \frac{c-b'}{a}$ . Si  $a$  es no constante, entonces por hipótesis inductiva tenemos que  $a$  toma todos los valores de  $F$ ; así que elegimos una asignación de las variables que haga  $a$  no nulo. Esta asignación de variables hace que  $a$  tome el valor  $a'$  y  $b$  tome el valor  $b'$ . Por último, elegimos  $x = \frac{c-b'}{a'}$ .

**Q. E. D.**

Ahora, para el problema SING tenemos:

**Teorema 6.4:** Si  $\mathbb{F}$  es un cuerpo y  $M$  una matriz  $cvuv$ , entonces el problema de decisión SING se encuentra en la clase de complejidad  $RP$ .

**Demostración:** El objetivo es replicar la demostración del Teorema 3.6. Hay que tomar en cuenta si llega a ocurrir que nuestro cuerpo  $\mathbb{F}$  no posee al menos  $2n$  elementos. Dado el caso podemos elegir un cuerpo extensión  $\mathbb{F}'$  tal que posea al menos  $2n$  elementos (en [7] demuestran que extender un cuerpo finito con  $q$  elementos a un cuerpo finito con  $q^n$  elementos se puede hacer en tiempo cuasi-polinomial).

Sea  $M(x_1, \dots, x_t)$  una matriz  $cvuv$  y definimos el polinomio  $p(x_1, \dots, x_t) = \det(M(x_1, \dots, x_t))$ . Aplicando los pasos de la demostración del Teorema 3.6, elegimos  $r$   $t$ -tuplas  $\mathbf{a}_1, \dots, \mathbf{a}_r$  y evaluamos el determinante de  $M$  en  $\mathbf{a}_i$ ,  $\det(M(\mathbf{a}_i))$ , para  $1 \leq i \leq r$ . Se nos presentarán tres posibles casos al hacer esto.

- Si de los  $r$  resultados obtenemos al menos dos distintos, entonces respondemos sí (debido a los Lemas 6.1 y 6.3,  $p$  toma el valor 0).
- Si todos los valores son iguales y son cero entonces respondemos sí.
- Si todos los valores son iguales y son no nulos entonces respondemos no.

La probabilidades asociadas a la respuesta ‘no’ se mantienen igual que en el Teorema 3.6. Además, usando el Corolario 6.2, tenemos que  $p$  no es idénticamente cero sobre  $\mathbb{F}'$ , si y solo si,  $p$  no es idénticamente cero sobre  $\mathbb{F}$ . De esta forma podemos computar SING sobre  $\mathbb{F}'$  en vez de  $\mathbb{F}$ . Como SING con  $S = \mathbb{F}'$  pertenece a  $RP$  entonces SING con  $S = \mathbb{F}$  también pertenece a  $RP$ .

**Q. E. D.**

Por último, mostraremos las cotas superiores para el problema de decisión MAXRANK sobre cuerpos para el caso cuando cada variable aparece una sola vez.

**Teorema 6.5:** Para todos los cuerpos  $\mathbb{F}$  y todas las matrices  $M$  que son  $cvuv$ , podemos computar  $\maxrank_{\mathbb{F}}(M)$  en tiempo polinomial aleatorio.

**Demostración:** Haremos una réplica de la demostración del Teorema 3.3. Sea  $M$  una matriz  $cvuv$  de tamaño  $n \times n$ . Si el cuerpo  $\mathbb{F}$  tiene al menos  $2n$  elementos, la demostración prosigue esencialmente sin cambios, con  $V$  cualquier sub-conjunto de  $\mathbb{F}$  de cardinal  $2n$ . Si  $\mathbb{F}$  no posee al menos  $2n$  elementos, elegimos un cuerpo extensión  $\mathbb{F}'$  con al menos  $2n$  elementos (en [7] demuestran que extender un cuerpo finito con  $q$  elementos a un cuerpo finito con  $q^n$  elementos

se puede hacer en tiempo cuasi-polinomial). Por el Corolario 6.2, tenemos que  $\det(M)$  (es decir, el polinomio multi-afín asociado a  $\det(M)$ ) no es idénticamente cero sobre  $\mathbb{F}'$ , si y solo si, no es idénticamente cero sobre  $\mathbb{F}$ . Así que podemos computar  $\text{maxrank}$  sobre  $\mathbb{F}'$  en vez de  $\mathbb{F}$ . Si  $\text{maxrank}_{\mathbb{F}'}(M)$  pertenece a  $RP$  entonces  $\text{maxrank}_{\mathbb{F}}(M)$  también pertenece a  $RP$ .

**Q. E. D.**

## Apéndice A

### Pseudo-Códigos de las demostraciones

En esta parte colocaremos los pseudo-códigos de los algoritmos usados en algunas de las demostraciones a lo largo del trabajo. Haremos uso del comando *print*, que permite mostrar en pantalla un mensaje al usuario; y del comando *return* que nos permite finalizar el algoritmo.

#### ALGORITMO Teorema 3.3.

Para este algoritmo nos tomaremos la libertad de usar la función [rank] (sin definirla explícitamente), la cual nos permite calcular el rango de una matriz.

#### ENTRADA:

- Matriz  $M \in \mathcal{M}(\mathbb{F} \cup \{x_1, \dots, x_t\})$ .
- Constante  $k$ .
- Cantidad de intentos  $r$ .

---

**Algoritmo 1** Verificar si  $\text{maxrank}(M) \geq k$ .

---

Se eligen aleatoriamente  $r$   $t$ -tuplas  $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r \in V^t$ ;

**for**  $i = 1 : 1 : r$  **do**

**if**  $\text{rank}(M(\mathbf{a}_i)) \geq k$  **then**  
        *print* "Sí";  
        *return*;  
    **end if**

**end for**

*print* "Con una probabilidad de al menos  $1 - \frac{1}{2^r}$  el  $\text{maxrank}(M)$  es menor que  $k$  .";

---

### ALGORITMO Teorema 3.6.

Para este algoritmo usaremos la función  $[\det]$ , la cual nos permite calcular el determinante de una matriz.

#### ENTRADA:

- Matriz  $M \in \mathcal{M}(\mathbb{F} \cup \{x_1, \dots, x_t\})$ .
- Cantidad de intentos  $r$ .

---

#### Algoritmo 2 Identificar si una matriz con entradas variables satisface SING.

---

Se eligen aleatoriamente  $r$   $t$ -tuplas  $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r \in V^t$ ;

**for**  $i = 1 : 1 : r$  **do**

$d_i = \det(M(\mathbf{a}_i))$ ;

**end for**

**if**  $d_1 = 0$  **then**

$print$  “ $M$  es singular.”;

$return$ ;

**end if**

**for**  $i = 1 : 1 : r$  **do**

**for**  $j = 1 : 1 : r$  **do**

**if**  $i \neq j$  **then**

**if**  $d_i \neq d_j$  **then**

$print$  “ $M$  es singular.”;

$return$ ;

**end if**

**end if**

**end for**

**end for**

$print$  “Con una probabilidad de al menos  $1 - \frac{1}{2^r}$   $M$  es no singular.”;

---

### ALGORITMO Proposición 4.1.

El siguiente código fue hecho en el lenguaje Java. Donde las dos primeras páginas muestran la creación de dos clases, la clase Arista y la clase Nodo; las cuales guardan la información necesaria para la construcción del grafo y de la matriz de adyacencia.

Luego, se muestra la clase principal, que utiliza métodos nativos del lenguaje (por ejemplo, replaceAll). Además de utilizar métodos propios creados (por ejemplo, calcular) para implementar el algoritmo representado en la Proposición 4.1.

#### ENTRADA:

- $F$  una fórmula.

---

Clase Arista.

---

```
public class Arista{
    String peso;
    Nodo siguiente;
    public String getPeso(){
        | return peso;
    }
    public void setPeso(String peso){
        | this.peso = peso;
    }
    public Nodo getSiguiente(){
        | return siguiente;
    }
    public void setSiguiente(Nodo siguiente){
        | this.siguiente = siguiente;
    }
    public Arista(String peso, Nodo siguiente){
        | this.peso = peso;
        | this.siguiente = siguiente;
    }
}
```

---

---

## Clase Nodo.

---

```
public class Nodo{
    private int valor;
    private List<Arista> hijos;
    public int getValor(){
        | return valor;
    }
    public void setValor(int valor){
        | this.valor = valor;
    }
    public List<Arista> getHijos(){
        | return hijos;
    }
    public void setHijos(List<Arista> hijos){
        | this.hijos = hijos;
    }
    public void setArista(Arista arista){
        | if hijos==null then
        | | hijos = new ArrayList<>();
        | end if
        | hijos.add(arista);
    }
    public Arista getArista(int pos){
        | return hijos.get(pos);
    }
}
```

---



---

**Algoritmo 3** Construcción del grafo  $G_{F'}$  y la matriz  $M$ 

---

```
public class Principal{
    private static Nodo inicial, ultimo;
    private static Arista aux;
    private static int countNodo, countHijos;
    private static int keepN[ ];
    private static String[ ][ ] M;
    private static List<Nodo> nodos;
    private static Nodo help;
    public Principal(){
        countHijos = 0;
        countNodo = 1;
        nodos = new ArrayList<>();
    }
    public void calcular(String formula){
        String convertida = formula.replaceAll("-", "+-");
        if(convertida.startsWith("+"))convertida = convertida.substring(1);
        String[ ] monomios = convertida.split("\\+");
        inicial = new Nodo();
        nodos.add(inicial);
        inicial.setValor(0);
        ultimo = new Nodo();
        nodos.add(ultimo);
        keepN = new int[monomios.length];
        for int i = 0; i < monomios.length; i++ do
            boolean isNegativo = monomios[i].contains("-");
            if isNegativo then
                generarGrafoResta(monomios[i]);
            else
                generarGrafoSuma(monomios[i]);
            end if
    }
}
```

---

---

```

    end for
    finalizarGrafo();
    llenarMatriz();
}
private static void generarGrafoSuma(String monomio){
    subGrafo(monomio);
    aux.setSiguiente(ultimo);
    keepN[countHijos] = help.getValor();
    countHijos++;
}
private static void generarGrafoResta(String monomio){
    subGrafo(monomio.substring(1));
    Nodo neg = new Nodo();
    nodos.add(neg);
    neg.setValor(countNodo);
    countNodo++;
    Arista arNeg = new Arista("1", null);
    arNeg.setSiguiente(ultimo);
    neg.setArista(arNeg);
    Arista lazoNeg = new Arista("1", null);
    neg.setArista(lazoNeg);
    lazoNeg.setSiguiente(neg);
    aux.setSiguiente(neg);
    help = neg;
    keepN[countHijos] = help.getValor();
    countHijos++;
}
private static void subGrafo(String monomio){
    String[] elementos = separar(monomio);

```

---

---

```
Arista p = new Arista(elementos[0],null);
Nodo ns = new Nodo();
nodos.add(ns);
ns.setValor(countNodo);
countNodo++;
Arista seg = new Arista("1", null);
ns.setArista(seg);
p.setSiguiente(ns);
inicial.setArista(p);
aux = ns.getArista(0);
help = ns;
Arista lazo = new Arista("1",null);
lazo.setSiguiente(ns);
ns.setArista(lazo);
for int i = 1; i < elementos.length; i++ do
    Arista r = new Arista(elementos[i],null);
    Nodo q = new Nodo();
    nodos.add(q);
    q.setValor(countNodo);
    countNodo++;
    Arista rr = new Arista("1",null);
    Nodo qq = new Nodo();
    nodos.add(qq);
    qq.setValor(countNodo);
    countNodo++;
    r.setSiguiente(qq);
    q.setArista(r);
    Arista lazoSeg = new Arista("1", null);
    lazoSeg.setSiguiente(q);
    q.setArista(lazoSeg);
```

---

---

```

    Arista lazoQ = new Arista("1", null);
    lazoQ.setSiguiente(qq);
    qq.setArista(rr);
    qq.setArista(lazoQ);
    aux.setSiguiente(q);
    help = qq;
    aux = qq.getArista(0);
  end for
}
private static String[] separar(String monomio){
  List<String> listaVariables = new ArrayList<>();
  String var = "";
  for char letra: monomio.toCharArray() do
    if var.equals("") then
      var+ = letra;
    else
      int ascii=(int)letra;
      if (ascii >= 65&&ascii <= 90)|| (ascii >= 97&&ascii <= 122) then
        listaVariables.add(var);
        var = String.valueOf(letra);
      else
        var+ = letra;
      end if
    end if
  end for
  listaVariables.add(var);
  return listaVariables.stream().toArray(String[]::new);
}
private static void finalizarGrafo(){
  ultimo.setValor(countNodo);
}

```

---

---

```
    Arista lazoFinal = new Arista("1", ultimo);
    ultimo.setArista(lazoFinal);
    Arista begEnd = new Arista("1", inicial);
    ultimo.setArista(begEnd);
}
private static void llenarMatriz(){
    int size = nodos.size();
    M = new String[size][size];
    for Nodo n : nodos do
        int fila = n.getValor();
        for Arista a : n.getHijos() do
            int columna = a.getSiguiente().getValor();
            M[fila][columna] = a.getPeso();
        end for
    end for
}
}
```

---

## Bibliografía

- [1] J. BUSS, G.S. FRANDBSEN, J.O. SHALLIT, The Computational Complexity Of Some Linear Algebra Problems: *BRICS Report Series 96-33* (1996). 1, 2, 30, 36
- [2] C. H. PAPANIMITRIOU, Computational Complexity, University of California, San Diego (1995). 1, 10
- [3] J. T. SCHWARTZ, Fast Probabilistic Algorithms for Verification of Polynomial Identities. *J. Assoc. Comput. Mach.* 27 (1980), 701-717. 22, 24
- [4] I. N. HERSTEIN, Álgebra Moderna: Grupos, Anillos, Campos, Teoría de Galois. 3
- [5] L. G. VALIANT, Completeness classes in algebra. In *Proc. Eleventh Ann. ACM Symp. Theor. Comput.*, pp. 249-261 (1979). 1
- [6] S. COOK, The complexity of theorem proving procedures. *Proceedings of the Third Annual ACM Symposium on Theory of Computing.* pp. 151-158 (1971). 2, 15
- [7] CATHERINE S. GREENHILL, Theoretical and Experimental Comparison of Efficiency of Finite Field Extensions. The Mathematical Institute, University of Oxford, U.K. (1994). 61
- [8] WILLIAM A. R. WEISS, An Introduction to Set Theory, (2008). 8
- [9] THANASES PHEIDAS, An effort to prove that the existential theory of  $\mathbb{Q}$  is undecidable. *Workshop on Hilbert's Tenth Problem: Relations with Arithmetic and Algebraic Geometry.* Ghent University, Belgium. (1999) 43