PESQUIMAT Revista de la Fac. CC. MM. de la
Universidad Nacional Mayor de San Marcos
Vol. X N°2, pp 19 - 44, LIMA - PERÚ. Noviembre 2007

Una técnica basada en el algoritmo de Euclides y sus aplicaciones a la Criptografía y Ecuaciones Diofánticas no lineales

Luis A. Cortés Vega[1]
lcortes@uantof.cl
Yolanda S. Santiago Ayala [2]
ysantiagoa@unmsm.edu.pe
Daniza E. Rojas Castro[1]
drojas@uantof.cl
Santiago C. Rojas Romero[2]
srojasr@unmsm.edu.pe

Resumen

El objetivo central de este trabajo es construir, a partir del algoritmo de Euclides, una matriz de algoritmos

$$\Phi_{\mathbb{B}} : \mathbb{N}^*_{m\times n} \to \mathbb{N}^*_{m\times n} \quad , \quad con \quad \Phi_{\mathbb{B}}(\mathbb{A}) = (\Phi_{b_{ij}}(a_{ij})),$$

donde $\mathbb{B} = (b_{ij})_{\substack{1\leq i\leq m \\ 1\leq j\leq n}}$ *es una matriz fija en* $\mathbb{N}^*_{m\times n}$. *La función* $\Phi_{\mathbb{B}}$ *es llamada* función matricial algorítmica. *Aquí mostramos sus propiedades y algunas de sus aplicaciones a la Criptografía y Ecuaciones diofánticas no lineales. De particular interés es el caso* $n = m = 1$. *En este sentido mostramos equivalencias entre* $\Phi_{\mathbb{B}}$ *y* la congruencia módulo p de Carl Friedrich Gauss.

Palabras Clave: *Función matricial algorítmica, Algoritmo de Euclides, Ecuaciones diofánticas no lineales, Decodificación y codificación de mensajes, Congruencia módulo p de Gauss.*

Abstract

The main objective of this work is to build, based on the Euclidean algorithm, a "matrix of algorithms"

$$\Phi_{\mathbb{B}} : \mathbb{N}^*_{m\times n} \to \mathbb{N}^*_{m\times n} \quad , \quad with \quad \Phi_{\mathbb{B}}(\mathbb{A}) = (\Phi_{b_{ij}}(a_{ij})),$$

[1]Universidad de Antofagasta, Departamento de Matemáticas, Facultad de Ciencias Básicas, Casilla 170, Chile.
[2]UNMSM, Facultad de Ciencias Matemáticas, Lima - Perú.

where $\mathbb{B} = (b_{ij})_{\substack{1\leq i\leq m \\ 1\leq j\leq n}}$ *is a fixed matrix on* $\mathbb{N}^*_{m\times n}$. *The function* $\Phi_{\mathbb{B}}$ *is called the algorithmic matrix function. Here we show its properties and some applications to Cryptography and nonlinear Diophantine equations. The case* $n = m = 1$ *has particular interest. On this way we show equivalences between* $\Phi_{\mathbb{B}}$ *and the Carl Friedrich Gauss's congruence module* p.

Keywords: *algorithmic matrix function, Euclidean algorithm, non-linear Diophantine equations, message codification and decoding, Gauss's congruence module* p .

1. Introducción y Motivación

Un representante del concepto algoritmo, el cual aparece ilustrado con frecuencia en la literatura matemática y constituye una herramienta fundamental, es el así llamado, **algoritmo de Euclides** (-330 a -227).

La palabra algoritmo tiene la connotación de un método preciso, que se sigue paso a paso, para resolver un problema en un número finito de "iteraciones".

Hoy en día, la comprensión de los más abstractos conceptos de las Matemáticas se puede facilitar, en alguna medida, con el soporte de programas algorítmicos.

Desde un punto de vista computacional, un algoritmo es un mecanísmo de cómputo programado, el cual debe ejecutar un número determinado de "iteraciones".

Un algoritmo, no siempre es representado por un programa computacional. Su implementación puede ser realizada por otros tipos de autómatas o en su defecto por el ser humano. Varios, y diferentes algoritmos, pueden realizar una misma tarea haciendo uso de un conjunto diferenciado de instrucciones, ejecutadas en un tiempo adecuado .

El concepto de algoritmo fué formalizado en 1936 por la **Máquina de Turing**, del matemático Alan Turing [12]. Estas ideas sobre algoritmos se convirtieron en los pilares fundamentales de la Computación Científica, ver [10].

La parte central del presente trabajo consiste en construir una técnica matemática algorítmica –con base en el algoritmo de Euclides– la que posteriormente aplicamos a la Criptografía y problemas de Teoría de Números. Aquí mostramos la riqueza e importancia de varios conceptos Matemáticos y Computacionales, entre los que se encuentran: el concepto de algoritmo computacional, el algoritmo de Euclides, el concepto de divisibilidad, el concepto de función isomorfa, el concepto de matriz y sus propiedades, así como el concepto de congruencia módulo p. En este contexto, ponemos también un especial énfasis al estudio de algunos problemas insertos en la teoría de las ecuaciones diofánticas no lineales.

El trabajo es estructurado en cuatro secciones y un apéndice. La primera de estas secciones está básicamente dedicada a la construcción de una matriz rectangular de orden $m\times n$, donde en cada una de sus entradas actúan ciertas funciones escalares $\phi_{b_{ij}}(\cdot)$, las que llamamos "*funciones escalares algorítmicas*". Para ser más precisos, aquí construímos una aplicación matricial de la forma

$$\Phi_{\mathbb{B}} : \mathbb{N}^*_{m\times n} \longrightarrow \mathbb{N}^*_{m\times n} \quad , \text{con } \Phi_{\mathbb{B}}(\mathbb{A}) = (\phi_{b_{ij}}(a_{ij}))$$

donde $\mathbb{A}$ es una matriz rectangular de orden $m \times n$, dada por

$$\mathbb{A} = (a_{ij}) = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n-1} & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n-1} & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn-1} & a_{mn} \end{pmatrix} \in \mathbb{N}^*_{m\times n}$$

y $\Phi_{\mathbb{B}}(\mathbb{A})$ es la matriz algorítmica

$$\Phi_{\mathbb{B}}(\mathbb{A}) \overset{\text{def}}{=} (\phi_{b_{ij}}(a_{ij})) = \begin{pmatrix} \phi_{b_{11}}(a_{11}) & \phi_{b_{12}}(a_{12}) & \dots & \phi_{b_{1n}}(a_{1n}) \\ \phi_{b_{21}}(a_{21}) & \phi_{b_{22}}(a_{22}) & \dots & \phi_{b_{2n}}(a_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{b_{m1}}(a_{m1}) & \phi_{b_{m2}}(a_{m2}) & \dots & \phi_{b_{mn}}(a_{mn}) \end{pmatrix} \in \mathbb{N}^*_{m\times n}$$

con $\phi_{b_{ij}} : \mathbb{N}^* \to \{0, 1, 2, \dots, b_{ij} - 1\}$ y $\mathbb{B} = (b_{ij})$ es una matriz fija en $\mathbb{N}_{m\times n}$ con todas sus entradas $b_{ij} \geq 2$, $1 \leq i \leq m$, $1 \leq j \leq n$. La función $\Phi_{\mathbb{B}}$ así construída, es llamada "*función matricial algorítmica*". Ahora bién, en la segunda sección del trabajo se muestran sus propiedades, teoremas y aplicaciones a la codificación y decodificación de mensajes. A continuaciòn, en la tercera sección del trabajo, aparecen algunas aplicaciones a ecuaciones diofánticas no lineales. En la cuarta sección abordamos el caso $n = m = 1$ y mostramos algunas equivalencias entre el concepto de función escalar algorítmica y el concepto de *congruencia módulo p de Carl Friedrich Gauss.* Finalizamos el trabajo anexando un apéndice en el cual proporcionamos un programa que computa el residuo de la división de dos números enteros positivos.

2. Construcción de la función matricial algorítmica

Comenzamos el trabajo revisando algunos conceptos y resultados que serán utilizados a lo largo del mismo.

El siguiente Teorema es uno de los resultados más importantes de la aritmética, [11].

Teorema 2.1. (*Algoritmo de Euclides*)

Dados dos enteros a y b, con $b > 0$, entonces existen únicos enteros q y r tales que

$$a = bq + r, \quad \text{con} \quad 0 \leq r < b.$$

Los números q y r reciben el nombre de cociente y resto de la división entera de a entre b.

La demostración de este Teorema es clásica, la damos con la finalidad de beneficiar la lectura del artículo.

Demostración.

Si b divide a a, el resultado es válido con $r = 0$, por lo que ahora consideramos sólo el caso en que b no divide a.

Sea $S = \{a - tb/t \in \mathbb{Z}, a - tb > 0\}$. Si $a > 0$ y $t = 0$, entonces $a \in S$ y S es no vacío. Si $a \leq 0$, sea $t = a - 1$, entonces $a - tb = a - (a-1)b = a(1-b) + b$, con $(1-b) \leq 0$, ya que $b \geq 1$. Así, $a - tb > 0$ y S es no vacío. Por lo tanto, para todo $a \in \mathbb{Z}$, S es un subconjunto no vacío de $\mathbb{Z}$. Por el principio del buen orden, el conjunto S debe tener un elemento mínimo r, tal que $0 < r = a - qb$, para algún $q \in \mathbb{Z}$. Si $r = b$, entonces $a = (q+1)b$ y b divide a, lo que contradice el hecho de que b no divide a. Ahora si $r > b$, entonces $r = b + c$, para algún $c \in \mathbb{Z}$ y $a - qb = r = b + c$, lo cual implica que $c = a - (q+1)b \in S$, lo que contradice que r sea el elemento mínimo del conjunto S. Por lo tanto, $r < b$. □

Una consecuencia de este Teorema es el Corolario 2.2 dado más abajo. Antes de enunciarlo, denotemos por $\mathbb{N}$, el conjunto de todos los números naturales y $\mathbb{N}^* = \mathbb{N} \cup \{0\}$; también, el símbolo $\times$ denotará el producto usual en $\mathbb{N}$.

Cabe mencionar aquí que si bien el Teorema 2.1 garantiza la existencia del resto r cuando dividimos un entero a (dividendo) entre b (divisor), no indica cómo se calcula este resto. Sólo con el objetivo de ayudar al lector incorporamos en el Apéndice del trabajo un programa que nos permite calcular el resto de una división; este programa es una adaptación de un algoritmo dado en el texto de R. Grimaldi, [13], p.p 217-218.

Corolario 2.2. Dados $a, b \in \mathbb{N}^*$ con $b \geq 2$, existen y son únicos los valores r_i, $0 \leq i \leq n$, $0 \leq r_i \leq b - 1$, tales que

$$a = r_0 + r_1 b + r_2 b^2 + \ldots + r_{n-1} b^{n-1} + r_n b^n$$

siendo n tal que $a < b^{n+1}$.

El Corolario 2.2 y el Teorema 2.1 dan pie a la definición de una aplicación sobre la cual esta basada toda la arquitectura del artículo.

Definición 2.3. Dados $a, b \in \mathbb{N}^*$ con $b \geq 2$, tales que,

$$a = r_0 + r_1 b + r_2 b^2 + \ldots + r_{n-1} b^{n-1} + r_n b^n,$$

con $0 \leq r_i \leq b - 1$, para todo $0 \leq i \leq n$, definimos una aplicación en $\mathbb{N}^*$, de la siguiente manera:

$$\phi_b : \mathbb{N}^* \to \mathbb{N}^*, \text{ tal que } \phi_b(a) = \begin{cases} a, & \text{si } 0 \leq a \leq b-1, \\ r_0, & \text{si } a \geq b. \end{cases}$$

Sólo para fijar ideas, ilustramos con algunos ejemplos cómo opera ϕ_b.

Ejemplo 2.4. A partir de la definición de ϕ_b tenemos:

(a) $\phi_3(9) = 0$, (b) $\phi_{17}(23) = 6$, (c) $\phi_8(25) = 1$,
(d) $\phi_{13}(5) = 5$, (e) $\phi_8(9) = 1$, (f) $\phi_5(\phi_8(21)) = 0$,
(g) $\phi_7(\phi_{15}(\phi_6(21))) = 3$.

Observación 2.5.

(i) Si en la Definición 2.3, $a \geq b$, entonces r_0 representa al resto de la división de a entre b, y por tanto, $0 \leq r_0 \leq b-1$.

(ii) De la Definición 2.3 podemos desprender el siguiente algoritmo :

Denotemos inicialmente η_0 por $\phi_{b_0}(a)$, donde $b_0 \geq 2$ y $a \geq b_0$, esto es $\eta_0 = \phi_{b_0}(a)$. Entonces, para $b_1 \geq 2$ tenemos

$$\phi_{b_1}(\eta_0) = \begin{cases} \eta_0, & \text{si } 0 \leq \eta_0 \leq b_1 - 1, \\ r_1, & \text{si } \eta_0 \geq b_1, \end{cases}$$

de donde

$$\phi_{b_1}(\phi_{b_0}(a)) = \begin{cases} \eta_0, & \text{si } 0 \leq \eta_0 \leq b_1 - 1, \\ r_1, & \text{si } \eta_0 \geq b_1. \end{cases}$$

Ahora, sea $\eta_1 = \phi_{b_1}(\phi_{b_0}(a))$ y $\eta_0 \geq b_1$, entonces para $b_2 \geq 2$, tenemos

$$\phi_{b_2}(\eta_1) = \begin{cases} \eta_1, & \text{si } 0 \leq \eta_1 \leq b_2 - 1, \\ r_2, & \text{si } \eta_1 \geq b_2, \end{cases}$$

de donde

$$\phi_{b_2}(\phi_{b_1}(\phi_{b_0}(a))) = \begin{cases} \eta_1, & \text{si } 0 \leq \eta_1 \leq b_2 - 1, \\ r_2, & \text{si } \eta_1 \geq b_2. \end{cases}$$

Continuando de esta manera, hasta el $k-1$ paso, con $\phi_{b_{k-1}}(\phi_{b_{k-2}}(\ldots \phi_{b_0}(a))) = \eta_{k-1}$, entonces para $b_k \geq 2$ y $\eta_{k-2} \geq b_{k-1}$, tenemos

$$\phi_{b_k}(\eta_{k-1}) = \begin{cases} \eta_{k-1}, & \text{si } 0 \leq \eta_{k-1} \leq b_k - 1, \\ r_k, & \text{si } \eta_{k-1} \geq b_k, \end{cases}$$

de donde

$$\eta_k = \phi_{b_k}(\phi_{b_k-1}(\phi_{b_{k-2}}(\ldots \phi_{b_0}(a)))) = \begin{cases} \eta_{k-1}, & \text{si } 0 \leq \eta_{k-1} \leq b_k - 1, \\ r_k, & \text{si } \eta_{k-1} \geq b_k. \end{cases}$$

Lo que denotaremos de aquí en adelante por

$$\eta_k = \phi_{b_k} \circ \phi_{b_k-1} \circ \phi_{b_{k-2}} \circ \ldots \circ \phi_{b_0}(a) = \begin{cases} \eta_{k-1}, & \text{si } 0 \leq \eta_{k-1} \leq b_k - 1, \\ r_k, & \text{si } \eta_{k-1} \geq b_k. \end{cases}$$

Note que este algoritmo se detiene en la k-ésima iteración si $0 \leq \eta_{k-1} < b_k$, y puede ser programado de manera similar al algoritmo dado en el Apéndice.

El siguiente ejemplo, nos ilustra el algoritmo anterior.

Ejemplo 2.6.
Note la evidencia del resultado $\phi_9(\phi_7(\phi_8(\phi_{24}(\phi_{62}(100))))) = 6$. En efecto, tene-mos

$$\begin{aligned}\phi_9(\phi_7(\phi_8(\phi_{24}(\phi_{62}(100))))) &= \phi_9(\phi_7(\phi_8(\phi_{24}(38))))\\ &= \phi_9(\phi_7(\phi_8(14))) = \phi_9(\phi_7(6)) = \phi_9(6) = 6.\end{aligned}$$

Conclusión:

$$\phi_9 \circ \phi_7 \circ \phi_8 \circ \phi_{24} \circ \phi_{62}(100) = 6.$$

Es claro que en el ejemplo tenemos: $b_0 = 62$, $b_1 = 24$, $b_2 = 8$, $b_3 = 7$ y $b_4 = 9$. Note además, que si $b_0 = b_1 = \ldots = b_k = b$, entonces el algorítmo se detiene en la primera iteración, ya que por la definición de ϕ_b, la fórmula $\phi_b(a) = \phi_b(\phi_b(a))$ siempre se verifica.

Observación 2.7.

(i) Dados $x, y \in \mathbb{N}^*$, $x \geq b, y \geq b$ tales que $x = y$, entonces $\phi_b(x) = \phi_b(y)$, esto hace de ϕ_b una función para cada $b \in \mathbb{N}^*$, con $b \geq 2$.

(ii) En este contexto, el dominio de la función ϕ_b para cada $b \in \mathbb{N}^*$, con $b \geq 2$ es el conjunto $\mathbb{N}^*$, y su imagen es el conjunto $\phi_b(\mathbb{N}^*) = \{0, 1, 2, \ldots, b-1\} \subset \mathbb{N}^*$.

(iii) La función ϕ_b al actuar sobre $a \in \mathbb{N}^*$, con $a \geq b$, entrega el resto r, de dividir a entre b.

(iv) La función ϕ_b , para cada $b \in \mathbb{N}^*$, con $a \geq b$, $b \geq 2$, **no es inyectiva** (ver Ejemplo 2.4 partes (c) y (e), más arriba).

Teorema 2.8. Dados a, b y $c \in \mathbb{N}^*$, con $b \geq 2$, $a \geq b$ y $c \geq b$ y sean además, r_{0a} y r_{0c} lo primeros 0-ésimos coeficientes que aparecen en la descomposición de a y c en la base b, respectivamente. Entonces ϕ_b verifica las siguientes propiedades:

(a) $\phi_b(0) = 0$,

(b) $\phi_b(db) = 0$, para todo $d \in \mathbb{N}^*$,

(c) $\phi_b(a) = \phi_b(\phi_b(a))$,

(d) $\phi_b(a + c) = \phi_b(\phi_b(a) + \phi_b(c))$,

(e) $\phi_b(a \times c) = \phi_b(\phi_b(a) \times \phi_b(c))$

(f) $\phi_b(a \times c) = \phi_b(a \times \phi_b(c))$

(g) $\phi_b(a \times c) = \phi_b(\phi_b(a) \times c)$

(h) $\phi_b(a + b) = \phi_b(a)$ ("periodicidad" de ϕ_b).

Observación 2.9.

Es fácil ver – a través del algoritmo de Euclides – que para todo $a \in \mathbb{N}^*$ tal que $a \geq b$, podemos caracterizar la función ϕ_b de la siguiente forma:

$$\phi_b(a) = r, \text{ si y sólo si, existe } q \in \mathbb{N}^*, \text{ tal que } a = r + bq, \text{ con } 0 \leq r \leq b-1.$$

Por ello y por la Observación 2.5, es que nombramos a la aplicación ϕ_b de *"función escalar algorítmica"*. La función escalar algorítmica, puede pensarse como un proceso iterativo, del cual obtenemos en k iteraciones, un número natural η_k, tal que $0 \leq \eta_k < b_k$.

A continuación nos valemos de la caracterización de la función escalar algorítmica ϕ_b, dada por la Definición 2.3 para demostrar el Teorema 2.8.

Demostración del Teorema 2.8

(a) Note que existe $q = 0 \in \mathbb{N}^*$, tal que $0 = 0 + 0b$, osea, $\phi_b(0) = 0$. También esta parte del Teorema 2.8, puede ser probada, usando la Definición 2.3.

(b) Note que existe $q = d \in \mathbb{N}^*$, tal que $db = 0 + db$, o sea, $\phi_b(db) = 0$.

(c) Sea $\phi_b(a) = r_{0a}$, entonces, $r_{0a} \leq b - 1$. Esto y de la Observación 2.5 implican que $\phi_b(r_{0a}) = r_{0a}$, es decir, $\phi_b(\phi_b(a)) = r_{0a} = \phi_b(a)$.

(d) Sean $r_{0a} = \phi_b(a)$ y $r_{0c} = \phi_b(c)$, entonces existen q_1 y $q_2 \in \mathbb{N}^*$, tales que $a = r_{0a} + q_1 b$ y $c = r_{0c} + q_2 b$, con $0 \leq r_{0a} \leq b-1$ y $0 \leq r_{0c} \leq b-1$. o sea, existe $q = q_1 + q_2 \in \mathbb{N}^*$, tal que

$$(a + c) = (r_{0a} + r_{0c}) + qb. \tag{2.1}$$

Por otro lado, sea $\phi_b(a + c) = r_{0(a+c)}$, entonces existe q_3, tal que

$$(a + c) = r_{0(a+c)} + q_3 b, \tag{2.2}$$

con $0 \leq r_{0(a+c)} \leq b - 1$. Ahora de (2.1) y (2.2) logramos encontrar un $q_4 = q - q_3 \in \mathbb{N}^*$, (ó $q_4 = q_3 - q \in \mathbb{N}^*$) tal que $(r_{0a} + r_{0b}) = r_{0(a+c)} + q_4 b$, o sea $\phi_b(r_{0a} + r_{0c}) = r_{0(a+c)}$. De donde

$$\phi_b(\phi_b(a) + \phi_b(c)) = \phi_b(a + c).$$

Esto finaliza la demostración del ítem (d).

(f)–(g) Sea $\phi_b(a) = r_{0a}$, entonces existe $q_1 \in \mathbb{N}^*$, tal que $a = r_{0a} + q_1 b$, con $0 \leq r_{0a} \leq b-1$. Ahora, para todo $z \in \mathbb{N}^*$ tenemos

$$\begin{aligned}
&\phi_b(z \times c) = \phi_b(\underbrace{c + c + c + \ldots + c}_{z- \text{ veces}}) = \phi_b(c + (c + c + \ldots + c)) \overset{(d)}{=} \\
&\overset{(d)}{=} \phi_b(\phi_b(c) + \phi_b(\underbrace{c + c + \ldots + c}_{z-1 \text{ veces}})) \overset{(d)}{=} \phi_b(\phi_b(c) + \phi_b(c) + \phi_b(\underbrace{c + c + \ldots + c}_{z-2 \text{ veces}})) \overset{(d)}{=} \\
&\overset{(d)}{=} \ldots \overset{(d)}{=} \phi_b(\underbrace{\phi_b(c) + \phi_b(c) + \ldots \phi_b(c)}_{z \text{ veces}}) = \phi_b(z\phi_b(c)).
\end{aligned} \tag{2.3}$$

Ahora, haciendo el cambio de variable $z = a$ en la ecuación (2.3), se sigue entonces que $\phi_b(a \times c) = \phi_b(a\phi_b(c))$. La demostración del ítem (g) es análoga.

(e) Sea $\phi_b(a) = r_{0a}$, entonces existe $q_1 \in \mathbb{N}^*$, tal que $a = r_{0a} + q_1 b$, con $0 \leq r_{0a} \leq b-1$. Usando elítem (f), probado anteriormente, tenemos

$$\begin{aligned} \phi_b(a \times c) &= \phi_b(a\phi_b(c)) = \phi_b((r_{0a} + q_1 b)\phi_b(c)) = \\ &= \phi_b(r_{0a}\phi_b(c) + q_1 b\phi_b(c)) \overset{(d)-(b)}{=} \phi_b(r_{0a}\phi_b(c)) \overset{(c)}{=} \phi_b(\phi_b(a)\phi_b(c)). \end{aligned}$$

Esto finaliza la demostración del ítem (e).

Para demostrar el ítem **(h)**, basta utilizar los ítems (d), (b) y (c). Por tanto, el Teorema 2.8 queda completamente probado. □

Corolario 2.10.

Sean $a_1, a_2, \ldots, a_n \in \mathbb{N}^*$, y sean p, $b \in \mathbb{N}^*$, tales que $b \geq 2$ y $p \geq 1$. Entonces:

(i) $\phi_b(\sum_{k=0}^{p} a_k) = \phi_b(\sum_{k=0}^{p} \phi_b(a_k))$,

(ii) $\phi_b(\prod_{k=o}^{p} a_k) = \phi_b(\prod_{k=o}^{p} \phi_b(a_k))$.

Demostración: Se desprende directamente del Teorema 2.8, ítems (c) y (d).

Por otro lado, resulta evidente el siguiente corolario

Corolario 2.11.

Sean a, p, $b \in \mathbb{N}^*$, tales que $b \geq 2$ y $p \geq 1$. Entonces:

$$\phi_b(a^p) = \phi_b([\phi_b(a)]^p).$$

Demostración: Se desprende del Corolario 2.10, ítem (ii).

Definición 2.12 (Función Matricial Algorítmica)

Dados $a_{ij}, b_{ij} \in \mathbb{N}^*$ con $b_{ij} \geq 2$, para todo $1 \leq i \leq m, 1 \leq j \leq n$. Consideramos ahora ϕ_{bij}, $1 \leq i \leq m, 1 \leq j \leq n$ funciones escalares algorítmicas de la forma

$$\phi_{bij} : \mathbb{N}^* \to \mathbb{N}^*, \text{ tal que } \phi_{b_{ij}}(a_{ij}) = \begin{cases} a_{ij}, & \text{si } 0 \leq a_{ij} \leq b_{ij} - 1, \\ r_{0_{ij}}, & \text{si } a_{ij} \geq b_{ij}. \end{cases}$$

A partir de ellas, construímos una aplicación matricial $\Phi_{\mathbb{B}}$:

$$\Phi_{\mathbb{B}} : \mathbb{N}^*_{m\times n} \to \mathbb{N}^*_{m\times n}, \text{ con } \Phi_{\mathbb{B}}(\mathbb{A}) = (\phi_{b_{ij}}(a_{ij})),$$

donde $\mathbb{A}$ es una matriz de orden $m \times n$ construída a partir de los a_{ij}:

$$\mathbb{A} = (a_{ij}) = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \ldots & a_{1n-1} & a_{1n} \\ a_{21} & a_{22} & a_{23} & \ldots & a_{2n-1} & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \ldots & a_{mn-1} & a_{mn} \end{pmatrix} \in \mathbb{N}^*_{m\times n},$$

con

$$\Phi_{\mathbb{B}}(\mathbb{A}) \stackrel{\text{def}}{=} (\phi_{b_{ij}}(a_{ij})) = \begin{pmatrix} \phi_{b_{11}}(a_{11}) & \phi_{b_{12}}(a_{12}) & \dots & \phi_{b_{1n}}(a_{1n}) \\ \phi_{b_{21}}(a_{21}) & \phi_{b_{22}}(a_{22}) & \dots & \phi_{b_{2n}}(a_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{b_{m1}}(a_{m1}) & \phi_{b_{m2}}(a_{m2}) & \dots & \phi_{b_{mn}}(a_{mn}) \end{pmatrix} \in \mathbb{N}^*_{m\times n}$$

Aquí, $\mathbb{B} = (b_{ij})$ es una "matriz de base" fija en $\mathbb{N}^*_{m\times n}$, con todas sus entradas $b_{ij} \geq 2$. La aplicación $\Phi_{\mathbb{B}}$, así construída, la llamamos *función matricial algorítmica*. Esta función matricial posee interesantes propiedades.

Teorema 2.13.

Dadas $\mathbb{A}, \mathbb{B}$ y $\mathbb{C}$ matrices en $\mathbb{N}^*_{m\times n}$, con $\mathbb{A} = (a_{ij})$, $\mathbb{B} = (b_{ij})$ y $\mathbb{C} = (c_{ij})$, tales que $b_{ij} \geq 2$, con $a_{ij} \geq b_{ij}$ y $c_{ij} \geq b_{ij}$. Entonces la función matricial algorítmica $\Phi_{\mathbb{B}}$, verifica las siguientes propiedades:

(a) $\Phi_{\mathbb{B}}(\mathbb{O}) = \mathbb{O}$,

(b) $\Phi_{\mathbb{B}}(d\mathbb{B}) = \mathbb{O}$, para todo $d \in \mathbb{N}^*$,

(c) $\Phi_{\mathbb{B}}(\mathbb{A}) = \Phi_{\mathbb{B}}(\Phi_{\mathbb{B}}(\mathbb{A}))$,

(d) $\Phi_{\mathbb{B}}(\mathbb{A} + \mathbb{C}) = \Phi_{\mathbb{B}}(\Phi_{\mathbb{B}}(\mathbb{A}) + \Phi_{\mathbb{B}}(\mathbb{C}))$,

(e) $\Phi_{\mathbb{B}}(d\mathbb{A}) = \Phi_{\mathbb{B}}(d\Phi_{\mathbb{B}}(\mathbb{A}))$,

(f) Dadas $\mathbb{B} = (b_{ij})$, $\mathbb{E} = (e_{ij})$, $\mathbb{F} = (f_{ij}) \in \mathbb{N}^*_{p\times p}$, con $b_{ij} \geq 2$. Sea $\mathbb{E} \cdot \mathbb{F} = \mathbb{D} = (d_{ij})$, así $d_{ij} = \sum_{k=1}^{p} e_{ik} f_{kj}$, $1 \leq i, j \leq p$, entonces

$$\Phi_{\mathbb{B}}(\mathbb{E} \cdot \mathbb{F}) = (\phi_{b_{ij}}(d_{ij})) = (\phi_{b_{ij}}(\sum_{k=1}^{p} \phi_{b_{ij}}(e_{ik}) \times \phi_{b_{ij}}(f_{kj}))).$$

(g) Sea $\mathbb{G} = (g_{ij}) \in \mathbb{N}^*_{p\times p}$, una matriz cuyas entradas g_{ij} son tales que $g_{ij} < b_{ij}$, con $b_{ij} \geq 2$, para todo $1 \leq i \leq m$, $1 \leq j \leq n$. Entonces,

$$\Phi_{\mathbb{B}}(\mathbb{G}) = \mathbb{G}.$$

Y en particular tenemos,

(h) $\Phi_{\mathbb{B}}(\mathbb{I}) = \mathbb{I}$, donde $\mathbb{I}$ denota a la matriz identidad en $\mathbb{N}^*_{m\times n}$,

(i) $\Phi_{\mathbb{B}}(\mathbb{A} + \mathbb{B}) = \Phi_{\mathbb{B}}(\mathbb{A})$ ("periocidad" de $\Phi_{\mathbb{B}}$).

Demostración:

Las demostraciones de **(a)**-**(g)** siguen inmediatamente del Teorema 2.8 y de la definición de $\Phi_{\mathbb{B}}$. La demostración del ítem **(h)** sigue de la definición de $\Phi_{\mathbb{B}}$, y del hecho que $b_{ij} \geq 2 > 0$, $b_{ij} \geq 2 > 1$, para todo $1 \leq i \leq m, 1 \leq j \leq n$. El ítem **(i)** sigue directamente del ìtem **(h)** del Teorema 2.8. □

Observación 2.14.

Es interesante observar que si $\mathbb{A} = (a_{ij})$, $\mathbb{B} = (b_{ij})$ y $\mathbb{C} = (c_{ij}) \in \mathbb{N}^*_{m\times n}$ son tales que $b_{ij} \geq 2$, $c_{ij} \geq 2$, para todo $1 \leq i \leq m, 1 \leq j \leq n$, entonces la siguiente operación matricial puede ser utilizada:

$$\Phi_{\mathbb{C}}(\Phi_{\mathbb{B}}(\mathbb{A})) \stackrel{\text{def}}{=} \begin{pmatrix} \phi_{c_{11}}(\phi_{b_{11}}(a_{11})) & \phi_{c_{12}}(\phi_{b_{12}}(a_{12})) & \cdots & \phi_{c_{1n}}(\phi_{b_{1n}}(a_{1n})) \\ \phi_{c_{21}}(\phi_{b_{21}}(a_{21})) & \phi_{c_{22}}(\phi_{b_{22}}(a_{22}) & \cdots & \phi_{c_{2n}}(\phi_{b_{2n}}(a_{2n})) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{c_{m1}}(\phi_{b_{m1}}(a_{m1})) & \phi_{c_{m2}}(\phi_{b_{m2}}(a_{m2})) & \cdots & \phi_{c_{mn}}(\phi_{b_{mn}}(a_{mn})) \end{pmatrix}$$

Ahora, usando esta identidad, podemos generalizar a funciones matriciales algorítmicas todo el algorítmo presentado en la Observación 2.5.

Con este fín, ilustramos esto con el siguiente ejemplo.

Ejemplo 2.15.

Sea $\mathbb{A} = (a_{ij}) \in \mathbb{N}^*_{2\times 3}$, la matriz dada por

$$\mathbb{A} = \begin{pmatrix} 21 & 79 & 45 \\ 49 & 5 & 39 \end{pmatrix} \in \mathbb{N}^*_{2\times 3}.$$

Sean $\mathbb{B} = (b_{ij})$ y $\mathbb{C} = (c_{ij}) \in \mathbb{N}^*_{2\times 3}$ matrices de base dadas por

$$\mathbb{B} = \begin{pmatrix} 5 & 12 & 23 \\ 12 & 4 & 35 \end{pmatrix}, \quad \mathbb{C} = \begin{pmatrix} 2 & 7 & 9 \\ 34 & 8 & 15 \end{pmatrix} \in \mathbb{N}^*_{2\times 3}.$$

Entonces

$$\begin{aligned} \Phi_{\mathbb{C}}(\Phi_{\mathbb{B}}(\mathbb{A})) &= \begin{pmatrix} \phi_2(\phi_5(21)) & \phi_7(\phi_{12}(79)) & \phi_9(\phi_{23}(45)) \\ \phi_{34}(\phi_{12}(49)) & \phi_8(\phi_4(5)) & \phi_{15}(\phi_{35}(39)) \end{pmatrix} \\ &= \begin{pmatrix} \phi_2(1) & \phi_7(7) & \phi_9(22) \\ \phi_{34}(1) & \phi_8(1) & \phi_{15}(4) \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & 4 \\ 1 & 1 & 4 \end{pmatrix}. \end{aligned}$$

3. Un algoritmo para codificar y decodificar Mensajes: Una aproximación

En la Criptografía –del griego kryptos (ocultar) y grafos (escribir), literalmente escritura oculta – existen una gran variedad de interesantes problemas relacionados con la codificación y decodificación de información. En su esencia, la criptografía es el arte o ciencia de cifrar y descifrar información **utilizando técnicas matemáticas** que hagan posible el intercambio de mensajes de manera que **sólo puedan ser leídos por las personas a quienes van dirigidas**, ver [2] y referencias allí contenidas.

Esta sección se construye con la ayuda de los conceptos e ideas concebidas en la sección anterior. Aquí, abordamos algunos problemas concernientes a la Criptografía, concretamente consideramos la codificación y decodificación del idioma Español (esto no debe por ningún motivo restringir el análisis). Así, el objetivo principal de esta sección es mostrar que, a través de la función matricial algorítmica, podemos codificar y decodificar mensajes.

Los autores de este trabajo pensamos que las técnicas y métodos aquí desarro-llados tienen una gran parte de originalidad, y pueden llegar a ser muy útiles al aplicarlos a otras áreas del conocimiento, como por ejemplo Genética, específicamente, en la traducción y lectura de secuencias del ADN humano.

Un punto importante a ser destacado en la sección – que pronto describiremos – es el concepto de "*Niveles de Codificación y de Decodificación*". También, desde un punto de vista didáctico, esta sección fué concebida autosuficiente. Es la sensación de todo el grupo de trabajo, que las ideas aquí expuestas puedan germinar y ser llevadas a otros ámbitos, como por ejemplo, la Didáctica de la Matemática, Matemáticas y Finanzas, Matemáticas Discretas, Teoría de Juegos, Computación e Informática, Inteligencia Artificial, Aritmética, Ingeniería, entre otras áreas del saber.

Definición 3.1. (Matriz de Lenguaje)

Denotemos por $S_{m\times n}$ el espacio de todas las matrices cuyas entradas contienen sólo símbolos provenientes del alfabeto Español y/o un símbolo superfluo, el que denotaremos por &. Los símbolos en las entradas de la matriz pueden tener o no un orden pre establecido.

Llamaremos a todas las matrices de $S_{m\times n}$ "matrices de lenguaje". Si $\mathbb{L} \in S_{m\times n}$, entonces:

$$\mathbb{L} = \begin{pmatrix} l_{11} & l_{12} & l_{13} & \dots & l_{1n-1} & l_{1n} \\ l_{21} & l_{22} & l_{23} & \dots & l_{2n-1} & l_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ l_{m1} & l_{m2} & l_{m3} & \dots & l_{mn-1} & l_{mn} \end{pmatrix}_{m\times n}.$$

Abajo, ilustramos la Definición 3.1, con algunos ejemplos.

Ejemplo 3.2.

$$\mathbb{L} = \begin{pmatrix} c & w & w & s \\ \& & q & p & m \\ t & y & u & z \end{pmatrix} \in S_{3\times 4}, \quad , \mathbb{G} = \begin{pmatrix} d & e & j & a & m & e \\ q & u & e & \& & t & e \\ c & u & e & n & t & e \end{pmatrix} \in S_{3\times 6}.$$

Definición 3.3. (Isomorfismo) Sea S el conjunto de todas las letras del alfabeto más un signo superfluo, todos ordenados de la siguiente forma:

$$S = \{\&, a, b, c, \dots, x, y, z\} = \{s_1, s_2, s_3, \dots, s_{26}, s_{27}, s_{28}\}.$$

Sea C el conjunto de los veintisiete primeros números naturales, más el cero, esto es:

$$C = \{0, 1, 2, \dots, 25, 26, 27\} = \{c_1, c_2, c_3, \dots, c_{26}, c_{27}, c_{28}\}.$$

A partir de estos conjuntos, definimos el siguiente isomorfismo:

$$f : S \to C, \text{ con } f(s_k) = c_k,\ 1 \leq k \leq 28.$$

Note, por ejemplo que: $f(s_5) = f(d) = 4$, $f(s_{21}) = f(s) = 20$, $f(s_1) = f(\&) = 0$. Note además que la función inversa de f, $g = f^{-1}$ actúa de la forma

$$g : C \to S, \text{ con } s_k = g(c_k),\ 1 \leq k \leq 28.$$

Lo que nos permite concluir que $d = g(4)$, $s = g(20)$ y $\& = g(0)$.

Observación 3.4.

(i) La definición 3.3 no es restrictiva, en el sentido de que siempre es posible usar otros símbolos.

(ii) A las letras del alfabeto conocidas, agregamos –con la idea de simplificar la lectura del mensaje– un nuevo símbolo "&", este símbolo, denotará un espacio en blanco entre cada palabra o letras, y formará parte del cifrado y/o el decifrado del mensaje.

(iii) Cabe hacer notar que la codificación y decodificación de mensajes, es utilizada con frecuencia en la contrucción de códigos de domínios públicos y privados.

Definición 3.5. (Matriz Transformada)
Denotaremos por $\mathbb{T} = (t_{ij})$ a la "matriz transformada" de una matriz de lenguaje $\mathbb{L}$, esta matriz constará de $m \times n$ caracteres todos numéricos t_{ij}, tales que $t_{ij} < 28$, para todo $1 \leq i \leq m$, $1 \leq j \leq n$.

Estos caracteres serán anexados a las entradas de la matriz $\mathbb{T}$, a través de la siguiente aplicación:

$$F : S_{m\times n} \to \mathbb{N}^*_{m\times n}, \text{ tal que } F(\mathbb{L}) = \mathbb{T}, \text{ con } f(l_{i,j}) = t_{i,j},\ 1 \leq i \leq m,\ 1 \leq j \leq n.$$

Aquí, f es el isomorfismo dado por la Definición 3.3.

Para ilustrar esto, usaremos la matriz del ejemplo 3.2. En efecto, note que la matriz de lenguaje

$$\mathbb{L} = \begin{pmatrix} c & w & w & s \\ \& & q & p & m \\ t & y & u & z \end{pmatrix}$$

tiene por transformada a la matriz

$$\mathbb{T} = F(\mathbb{L}) = \begin{pmatrix} 3 & 24 & 24 & 20 \\ 0 & 18 & 17 & 13 \\ 21 & 26 & 22 & 27 \end{pmatrix}.$$

Definición 3.6. Llamaremos "matriz de retorno" a la matriz que se obtiene de la aplicación

$$R : N^*_{m\times n} \to \mathbb{S}_{m\times n}, \text{ con } R(\mathbb{T}) = (g(t_{i,j})),\ 1 \leq i \leq m, 1 \leq j \leq n,$$

donde g es la función inversa del isomorfismo f dado en la Definición 3.3. Para ilustrar esto, usaremos el ejemplo anterior, así tenemos que:

$$R(\mathbb{T}) = R\left(\begin{pmatrix} 3 & 24 & 24 & 20 \\ 0 & 18 & 17 & 13 \\ 21 & 26 & 22 & 27 \end{pmatrix}\right) = \begin{pmatrix} c & w & w & s \\ \& & q & p & m \\ t & y & u & z \end{pmatrix}.$$

Ahora, sea $\mathbb{L}_0 \in \mathbb{S}_{m\times n}$ una matriz de lenguaje; en esta matriz supondremos que se encontrará incorporado el mensaje que queremos codificar o decodificar. Sea también $\mathbb{C}_0 = (q_{ij}) = F(\mathbb{L}) \in \mathbb{N}^*_{m\times n}$, una matriz, a la que llamaremos *"matriz fuente"*; esta matriz tiene la particularidad de ser la transformada de una matriz de lenguaje $\mathbb{L}_0 \in \mathbb{S}_{m\times n}$, por tanto, contiene en todas sus entradas caracteres numéricos $q_{ij} < 28$, $1 \leq i \leq m$, $1 \leq j \leq n$. También, denotamos por $\mathbb{C}_p = (p_{ij}) \in \mathbb{N}^*_{m\times n}$, a la matriz que llamaremos de *"matriz de código personal"*; esta matriz tiene la particularidad de ser asignada a la persona que codificará el mensaje oculto en $\mathbb{L}_0$. Ahora estamos preparados para dar la siguiente definición.

Definición 3.7. Sean $\mathbb{L}_0 \in \mathbb{S}_{m\times n}$, $\mathbb{C}_0 \in \mathbb{N}^*_{m\times n}$ las matrices caracterizadas arriba. Sean $\mathbb{B} \in \mathbb{N}^*_{m\times n}$ una matriz de base de la forma

$$\mathbb{B} = \begin{pmatrix} 28 & 28 & 28 & \dots & 28 & 28 \\ 28 & 28 & 28 & \dots & 28 & 28 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 28 & 28 & 28 & \dots & 28 & 28 \end{pmatrix}_{m\times n}$$

y $\mathbb{C}_p^{(1)}, \mathbb{C}_p^{(2)}, \mathbb{C}_p^{(3)}, \dots, \mathbb{C}_p^{(k)}$ una sucesión de k matrices de código personal, con $\mathbb{C}_p^{(k)} = (c_{ij}^{(k)}) \in \mathbb{N}^*_{m\times n}$. Llamamos al nivel en donde se computa la matriz $\mathbb{C}_k$, resultante de la operación

$$\Phi_{\mathbb{B}}\left(\sum_{m=1}^{k} \mathbb{C}_p^{(m)} + \mathbb{C}_0\right) = \mathbb{C}_k,$$

como **$k-$ésimo nivel de codificación de la matriz $\mathbb{C}_0$ usando el código personal $\mathbb{C}_p^{(k)}$.**

Esta definición surge del análisis del siguiente esquema de codificación de la matriz $\mathbb{C}_0$. Para ello utilizamos las matrices de código personal $\mathbb{C}_p^{(m)}$, $m = 1, 2, \dots, k$ y la función matricial algorítmica $\Phi_{\mathbb{B}}$. Por ejemplo, si estamos en el k-ésimo nivel de codificación de la matriz $\mathbb{C}_0$, el esquema toma la forma:

$$\underbrace{\boxed{\underset{\text{Datos para el } k\text{-ésimo nivel}}{\mathbb{L}_0, \mathbb{C}_0, \left\{\mathbb{C}_p^{(m)}\right\}_{m=1}^{k}, \mathbb{B}}} \rightrightarrows \boxed{\underset{\text{Algorítmo en el } k\text{-ésimo nivel}}{\Phi_{\mathbb{B}}\left(\sum_{m=1}^{k} \mathbb{C}_p^{(m)} + \mathbb{C}_0\right)}} \rightrightarrows \boxed{\underset{\text{Salida en el } k\text{-ésimo nivel}}{\mathbb{C}_k,\ R(\mathbb{C}_k) = \mathbb{L}_k}}}_{\text{Repetimos el proceso si queremos volver a codificar } \mathbb{C}_0}$$

En otras palabras,
Datos iniciales: $\mathbb{L}_0 \in \mathbb{S}_{m\times n}$, $\mathbb{C}_0 \in \mathbb{N}^*_{m\times n}$, $\mathbb{C}_p^{(1)} \in \mathbb{N}^*_{m\times n}$, y $\mathbb{B} \in \mathbb{N}^*_{m\times n}$. Entonces

$$\left|\left[\begin{array}{l} \Phi_{\mathbb{B}}(\mathbb{C}_p^{(1)} + \mathbb{C}_0) = \mathbb{C}_1 \text{ es el primer nivel de codificación de } \mathbb{C}_0. \\ \text{Hacemos } R(\mathbb{C}_1) = \mathbb{L}_1 \end{array}\right.\right.$$

Datos iniciales: $\mathbb{L}_1 \in \mathbb{S}_{m\times n}$, $\mathbb{C}_1 \in \mathbb{N}^*_{m\times n}$, $\mathbb{C}_p^{(2)} \in \mathbb{N}^*_{m\times n}$, y $\mathbb{B} \in \mathbb{N}^*_{m\times n}$. Entonces

$$\left|\left[\begin{array}{l} \Phi_{\mathbb{B}}(\mathbb{C}_p^{(2)} + \mathbb{C}_1) = \mathbb{C}_2 \text{ es el segundo nivel de codificación de } \mathbb{C}_0. \\ \text{Hacemos } R(\mathbb{C}_2) = \mathbb{L}_2 \end{array}\right.\right.$$

Datos iniciales: $\mathbb{L}_2 \in \mathbb{S}_{m\times n}$, $\mathbb{C}_2 \in \mathbb{N}^*_{m\times n}$, $\mathbb{C}_p^{(3)} \in \mathbb{N}^*_{m\times n}$, y $\mathbb{B} \in \mathbb{N}^*_{m\times n}$. Entonces

$$\left|\left[\begin{array}{l} \Phi_{\mathbb{B}}(\mathbb{C}_p^{(3)} + \mathbb{C}_2) = \mathbb{C}_3 \text{ es el tercer nivel de codificación de } \mathbb{C}_0. \\ \text{Hacemos } R(\mathbb{C}_3) = \mathbb{L}_3, \end{array}\right.\right.$$

y así sucesivamente.
Note que después de $k-1$ niveles de codificación de la matriz $\mathbb{C}_0$, tenemos

Datos iniciales: $\mathbb{L}_{k-1} \in \mathbb{S}_{m\times n}$, $\mathbb{C}_{k-1} \in \mathbb{N}^*_{m\times n}$, $\mathbb{C}_p^{(k)} \in \mathbb{N}^*_{m\times n}$, y $\mathbb{B} \in \mathbb{N}^*_{m\times n}$. Entonces

$$\left|\left[\begin{array}{l} \Phi_{\mathbb{B}}(\mathbb{C}_p^{(k)} + \mathbb{C}_{k-1}) = \mathbb{C}_k \text{ es el } k-\text{ésimo nivel de codificación de } \mathbb{C}_0. \\ \text{Hacemos } R(\mathbb{C}_k) = \mathbb{L}_k \end{array}\right.\right.$$

Note además que al $k-$ésimo nivel de codificación, obtenemos:

$$\begin{aligned} \mathbb{C}_k &= \Phi_{\mathbb{B}}(\mathbb{C}_p^{(k)} + \mathbb{C}_{k-1}) = \Phi_{\mathbb{B}}(\mathbb{C}(_p^{(k)} + \underbrace{\Phi_{\mathbb{B}}(\mathbb{C}_p^{(k-1)} + \mathbb{C}_{k-2})}_{\mathbb{C}_{k-1}}) \\ &= \Phi_{\mathbb{B}}(\mathbb{C}_p^{(k)} + \Phi_{\mathbb{B}}(\mathbb{C}_p^{(k-1)} + \underbrace{\Phi_{\mathbb{B}}(\mathbb{C}_p^{(k-2)} + \mathbb{C}_{k-3})}_{\mathbb{C}_{k-2}})) \\ &= \ldots \overset{\text{Teo. 3.13}}{=} \Phi_{\mathbb{B}}(\underbrace{\mathbb{C}_p^{(k)} + \mathbb{C}_p^{(k-1)} + \mathbb{C}_p^{(k-2)} + \ldots + \mathbb{C}_p^{(1)}}_{k-\text{sumandos}} + \mathbb{C}_0) \\ &= \Phi_{\mathbb{B}}(\sum_{m=1}^{k} \mathbb{C}_p^{(m)} + \mathbb{C}_0). \end{aligned}$$

Este es el hecho que ha motivado la Definición 3.7.
En lo que sigue, denotaremos por $[x]_{pe}$ la parte entera de un número real x, es decir,

$$[x]_{pe} = máx\,\{k \in \mathbb{Z}, \text{ tal que } k \leq x\}.$$

El siguiente Teorema es fundamental en esta sección, ya que nos entrega una forma de decodificar una matriz $\mathbb{C}_k$ codificada en el $k-$ésimo nivel. Sin pérdida de generalidad, aquí sólo damos el Teorema al primer nivel de codificación.

Teorema 3.8. Dadas $\mathbb{C}_p^{(1)} = (c_{ij}^{(1)}) \in \mathbb{N}_{m\times n}^*$, matriz de código personal, y $\mathbb{B} \in \mathbb{N}_{m\times n}^*$ una matriz de base del tipo

$$\mathbb{B} = \begin{pmatrix} 28 & 28 & 28 & \dots & 28 & 28 \\ 28 & 28 & 28 & \dots & 28 & 28 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 28 & 28 & 28 & \dots & 28 & 28 \end{pmatrix}_{m\times n},$$

$\mathbb{L}_0 \in \mathbb{S}_{m\times n}$ una matriz de lenguaje, y $\mathbb{C}_d^{(1)} = d\mathbb{B} - \mathbb{C}_p^{(1)} \in \mathbb{N}_{m\times n}^*$ una matriz de "decodificación", con

$$d = \left[\frac{\text{máx}\left\{ c_{ij}^{(1)} : 1 \le i \le m;\ 1 \le j \le n \right\}}{28} + 1 \right]_{pe}.$$

Tome inicialmente $\mathbb{C}_0 = F(\mathbb{L}_0) \in \mathbb{N}_{m\times n}^*$, tal que

$$\begin{cases} \Phi_{\mathbb{B}}(\mathbb{C}_p^{(1)} + \mathbb{C}_0) = \mathbb{C}_1, \\ R(\mathbb{C}_1) = \mathbb{L}_1 \in \mathbb{S}_{m\times n}. \end{cases}$$

Entonces

$$\begin{cases} \Phi_{\mathbb{B}}(\mathbb{C}_d^{(1)} + \mathbb{C}_1) = \mathbb{C}_0, \\ R(\mathbb{C}_0) = \mathbb{L}_0 \in \mathbb{S}_{m\times n}. \end{cases} \tag{3.1}$$

Observación 3.9.

(i) No olvidemos que en la identidad (3.1), la matriz $\mathbb{C}_1 = (r_{ij}) \in \mathbb{N}_{m\times n}^*$ con $r_{ij} < 28$, $1 \le i \le m$, $1 \le j \le n$, representa a la matriz resultante de la codificación del mensaje contenido en $\mathbb{L}_0$, a través de $\Phi_{\mathbb{B}}$ usando la matriz de código personal $\mathbb{C}_p^{(1)} \in \mathbb{N}_{m\times n}^*$. La matriz $\mathbb{C}_d^{(1)}$, al interior de la identidad (3.1) induce la decodificación, a través de $\Phi_{\mathbb{B}}$ de la matriz $\mathbb{C}_1$. Lo que asegura el Teorema 3.8, es que con esa elección de $\mathbb{C}_d^{(1)}$ regresamos a la matriz $\mathbb{C}_0$.

(ii) El Teorema 3.8 nos da una técnica matemática algorítmica (la función matricial algorítmica), basada en el algorítmo de Euclides, la cual permite decodificar un mensaje codificado. Esto se lleva a cabo mediante la función algorítmica matricial $\Phi_{\mathbb{B}}$. En otras palabras, lo que es posible codificar vía $\Phi_{\mathbb{B}}$ usando $\mathbb{C}_p^{(1)}$, puede ser decodificado vía $\Phi_{\mathbb{B}}$ usando $\mathbb{C}_d^{(1)}$, y recíprocamente.

Demostración del Teorema 3.8. Suponga que el primer nivel de codificación de la

matriz $\mathbb{C}_0$ fué realizado, es decir, la identidad $\Phi_{\mathbb{B}}(\mathbb{C}_p^{(1)} + \mathbb{C}_0) = \mathbb{C}_1$ es válida, entonces

$$\Phi_{\mathbb{B}}(\mathbb{C}_1 + \mathbb{C}_d^{(1)}) \overset{Hip}{=} \Phi_{\mathbb{B}}(\underbrace{\Phi_{\mathbb{B}}(\mathbb{C}_p^{(1)} + \mathbb{C}_0)}_{\mathbb{C}_1} + \mathbb{C}_d^{(1)})$$

$$\overset{(d)}{=} \Phi_{\mathbb{B}}(\Phi_{\mathbb{B}}(\Phi_{\mathbb{B}}(\mathbb{C}_0 + \mathbb{C}_p^{(1)})) + \Phi_{\mathbb{B}}(\mathbb{C}_d^{(1)}))$$

$$\overset{(d)-(g)}{=} \Phi_{\mathbb{B}}(\Phi_{\mathbb{B}}(\mathbb{C}_0 + \mathbb{C}_p^{(1)}) + \Phi_{\mathbb{B}}(\mathbb{C}_d^{(1)})).$$

De donde,

$$\begin{aligned}
\Phi_{\mathbb{B}}(\mathbb{C}_1 + \mathbb{C}_d^{(1)}) &= \Phi_{\mathbb{B}}((\mathbb{C}_0 + \mathbb{C}_p^{(1)}) + \mathbb{C}_d^{(1)}) \\
&= \Phi_{\mathbb{B}}(\mathbb{C}_0 + \mathbb{C}_p^{(1)} + d\mathbb{B} - \mathbb{C}_p^{(1)}) \\
&= \Phi_{\mathbb{B}}(\mathbb{C}_0 + d\mathbb{B} + \mathbb{O}).
\end{aligned}$$

Luego,

$$\begin{aligned}
\Phi_{\mathbb{B}}(\mathbb{C}_1 + \mathbb{C}_d^{(1)}) &= \Phi_{\mathbb{B}}(\mathbb{C}_0 + d\mathbb{B}) \\
&\overset{(d)}{=} \Phi_{\mathbb{B}}(\Phi_{\mathbb{B}}(\mathbb{C}_0) + \Phi_{\mathbb{B}}(d\mathbb{B})) \\
&\overset{(b)}{=} \Phi_{\mathbb{B}}(\Phi_{\mathbb{B}}(\mathbb{C}_0) + \mathbb{O}) \\
&= \Phi_{\mathbb{B}}(\Phi_{\mathbb{B}}(\mathbb{C}_0)).
\end{aligned}$$

Y finalmente,

$$\Phi_{\mathbb{B}}(\mathbb{C}_1 + \mathbb{C}_d^{(1)}) = \Phi_{\mathbb{B}}(\Phi_{\mathbb{B}}(\mathbb{C}_0)) \overset{(c)}{=} \Phi_{\mathbb{B}}(\mathbb{C}_0) \overset{(g)}{=} \mathbb{C}_0.$$

Esto último, gracias al ítem (g) del Teorema 2.13, ya que las entradas de la matriz $\mathbb{C}_0 = (q_{ij}) = F(\mathbb{L}_0) \in \mathbb{N}^*_{m\times n}$ son todas $q_{ij} < 28$, $1 \leq i \leq m, 1 \leq j \leq n$. Así, finaliza la demostración del Teorema 3.8. □

Para intentar leer un mensaje codificado, debemos antes que todo realizar su decodificación; la matriz resultante de la decodificación tendría todas sus entradas en $\mathbb{N}^*$, para intentar leer esta matriz y por ende el mensaje, debemos aplicar a esta última, la función de retorno R. Ilustramos todas estas ideas a través del siguiente ejemplo.

Ejemplo 3.10. Supongamos que un colega desde Medellin (Colombia) nos envía, a través de su e-mail, un mensaje codificado el cual se encuentra contenido en la matriz de lenguaje

$$\mathbb{L} = \begin{pmatrix} i & a & k & x & f & o & c & n & d & e & z & r & z & n \\ v & \& & l & z & o & z & c & p & u & l & r & y & p & y \end{pmatrix} \in S_{2\times 14}.$$

Nuestro colega, nos dice que el mensaje ha sido codificado vía $\Phi_{\mathbb{B}}$, y que debería ser leído, es decir, decodificado, recién en el tercer nivel de decodificación; y que cada nivel de decodificación que realicemos deberíamos utilizar una matriz de código personal del tipo:

$$\mathbb{C}_p = (c_{ij}^{(p)}) = \begin{pmatrix} 19 & 23 & 9 & 8 & 2 & 3 & 4 & 13 & 17 & 20 & 21 & 25 & 37 & 14 \\ 2 & 4 & 7 & 9 & 13 & 21 & 22 & 24 & 29 & 32 & 34 & 12 & 17 & 15 \end{pmatrix} \in \mathbb{N}^*_{2\times 14}.$$

Esta información se nos hace llegar a través de Internet. Por tanto, si queremos leer el mensaje, deberíamos ser capaces de construir la matriz $\mathbb{C}_1$ y de allí $R(\mathbb{C}_1)$.

Si este es el objetivo, un primer paso debería ser determinar la transformada de $\mathbb{L} = \mathbb{L}_4$ vía F. Si hacemos esto, obtenemos la matriz de codificación $F(\mathbb{L}_4) = \mathbb{C}_4$, y por tanto

$$\mathbb{C}_4 = \begin{pmatrix} 9 & 1 & 11 & 25 & 6 & 16 & 3 & 14 & 4 & 5 & 27 & 19 & 27 & 14 \\ 23 & 0 & 12 & 27 & 16 & 27 & 3 & 17 & 22 & 12 & 19 & 26 & 17 & 26 \end{pmatrix} \in \mathbb{N}^*_{2\times 14}.$$

Para la decodificación del mensaje inicial, usaremos en adelante repetidas veces el Teorema 3.8. En efecto, sea $\mathbb{B} \in \mathbb{N}^*_{2\times 14}$, una matriz de base dada por

$$\mathbb{B} = \begin{pmatrix} 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \end{pmatrix}_{2\times 14},$$

la matriz de decodificado $\mathbb{C}_d$, se calcula usando la fórmula $\mathbb{C}_d = d\mathbb{B} - \mathbb{C}_p$, con

$$d = \left[\frac{\text{máx}\left\{ c_{ij}^{(p)} : 1 \le i \le 2;\ 1 \le j \le 14 \right\}}{28} + 1 \right]_{pe} = \left[\frac{37}{28} + 1 \right]_{pe} = [1{,}3214 + 1]_{pe} = 2 \quad (1)$$

así $\mathbb{C}_d \in \mathbb{N}^*_{2\times 14}$, y toma la forma:

$$\mathbb{C}_d = \begin{pmatrix} 37 & 33 & 47 & 48 & 54 & 53 & 52 & 43 & 39 & 36 & 35 & 31 & 19 & 42 \\ 54 & 52 & 49 & 47 & 43 & 35 & 34 & 32 & 27 & 24 & 22 & 44 & 39 & 41 \end{pmatrix}.$$

De donde obtenemos,

$$\Phi_{\mathbb{B}}(\mathbb{C}_4 + \mathbb{C}_d) = \mathbb{C}_3 = \begin{pmatrix} 18 & 6 & 2 & 17 & 4 & 13 & 27 & 1 & 15 & 13 & 6 & 22 & 18 & 0 \\ 21 & 24 & 5 & 18 & 3 & 6 & 9 & 21 & 21 & 8 & 13 & 14 & 0 & 11 \end{pmatrix},$$

esto induce la matriz de lenguaje

$$R(\mathbb{C}_3) = \mathbb{L}_3 = \begin{pmatrix} q & f & b & p & d & m & z & a & \tilde{n} & m & f & u & q & \& \\ t & w & e & q & c & f & i & t & t & h & m & n & \& & k \end{pmatrix}.$$

Es evidente que el mensaje oculto en $\mathbb{L}_3$, al primer nivel de decodificación, no es decifrable en nuestro lenguaje Español, por tanto, debemos avanzar a un segundo nivel de decodificación, para esto, de nuevo consideramos la matriz:

$$\mathbb{C}_3 = F(\mathbb{L}_3) = \begin{pmatrix} 18 & 6 & 2 & 17 & 4 & 13 & 27 & 1 & 15 & 13 & 6 & 22 & 18 & 0 \\ 21 & 24 & 5 & 18 & 3 & 6 & 9 & 21 & 21 & 8 & 13 & 14 & 0 & 11 \end{pmatrix}$$

Ahora, usando nuevamente la matriz de decodificado $\mathbb{C}_d$,

$$\mathbb{C}_d = \begin{pmatrix} 37 & 33 & 47 & 48 & 54 & 53 & 52 & 43 & 39 & 36 & 35 & 31 & 19 & 42 \\ 54 & 52 & 49 & 47 & 43 & 35 & 34 & 32 & 27 & 24 & 22 & 44 & 39 & 41 \end{pmatrix}$$

obtenemos,

$$\Phi_{\mathbb{B}}(\mathbb{C}_3 + \mathbb{C}_d) = \mathbb{C}_2 = \begin{pmatrix} 27 & 11 & 21 & 9 & 2 & 10 & 23 & 16 & 26 & 21 & 13 & 25 & 9 & 14 \\ 19 & 20 & 26 & 9 & 18 & 13 & 15 & 25 & 20 & 4 & 7 & 2 & 11 & 24 \end{pmatrix}.$$

Por tanto,

$$R(\mathbb{C}_2) = \mathbb{L}_2 = \begin{pmatrix} z & k & t & i & b & j & v & o & y & t & m & x & i & n \\ r & s & y & i & q & m & \tilde{n} & x & s & d & g & b & k & w \end{pmatrix}.$$

De nuevo observamos que el mensaje oculto en $\mathbb{L}_2$, al segundo nivel de deco-dificación, sigue estando impreciso, esto nos conduce a ejecutar un tercer nivel de decodificación. Para esto determinamos una tercera matriz

$$\mathbb{C}_2 = F(\mathbb{L}_2) = \begin{pmatrix} 27 & 11 & 21 & 9 & 2 & 10 & 23 & 16 & 26 & 21 & 13 & 25 & 9 & 14 \\ 19 & 20 & 26 & 9 & 18 & 13 & 15 & 25 & 20 & 4 & 7 & 2 & 11 & 24 \end{pmatrix}$$

y usando nuevamente la matriz de decodificado

$$\mathbb{C}_d = \begin{pmatrix} 37 & 33 & 47 & 48 & 54 & 53 & 52 & 43 & 39 & 36 & 35 & 31 & 19 & 42 \\ 54 & 52 & 49 & 47 & 43 & 35 & 34 & 32 & 27 & 24 & 22 & 44 & 39 & 41 \end{pmatrix}$$

obtenemos,

$$\Phi_{\mathbb{B}}(\mathbb{C}_2 + \mathbb{C}_d) = \mathbb{C}_1 = \begin{pmatrix} 8 & 16 & 12 & 1 & 0 & 7 & 19 & 3 & 9 & 1 & 20 & 0 & 0 & 0 \\ 17 & 16 & 19 & 0 & 5 & 20 & 21 & 1 & 19 & 0 & 1 & 18 & 22 & 9 \end{pmatrix}.$$

De esta operación, surge una nueva matriz de lenguaje:

$$R(\mathbb{C}_1) = \mathbb{L}_1 = \begin{pmatrix} h & o & l & a & \& & g & r & c & i & a & s & \& & \& & \& \\ p & o & r & \& & e & s & t & a & r & \& & a & q & u & i \end{pmatrix}.$$

De aquí, el mensaje envíado por nuestro colega es:

HOLA GRCIAS POR ESTAR AQUI.

Por tanto, en este nivel el proceso debe finalizar, y deberíamos ser capaces de intuir que el mensaje enviado por nuestro colega es:

HOLA, GRACIAS POR ESTAR AQUI.

Observación 3.11.

(i) Cabe hacer notar que la matriz de código personal $\mathbb{C}_p$, dada en nuestro ejemplo, puede variar en cada nivel de decodificación o codificación. Nosotros aquí en cada nivel la consideramos constante solo para simplicar los cálculos.

(ii) También – como intentaremos mostrar en un próximo trabajo – la función matricial algorítmica $\Phi_{\mathbb{B}}$ tiene otras aplicaciones, una de ellas tiene relación con la posibilidad de codificar y decodificar un mensaje en forma parcial.

4. Aplicaciones de la función escalar algorítmica a ecuaciones Diofánticas no lineales: Un mundo de conjeturas

La expresión "Ecuación diofántica" se debe a Diofantos de Alejandría (255 A.D.), uno de los grandes matemáticos de la civilización Griega. El fué el primero en iniciar un sistemático estudio de ecuaciones cuya resolución queda en el ámbito de los enteros positivos. El obtiene los primeros grandes logros en este campo. Diofantos escribió en esta línea de pensamiento a lo largo de su vida, tres importantes trabajos, siendo uno de lo más importantes el que conocemos hoy en día como "Aritmética". En esta obra, Diofantos entrega soluciones enteras a ecuaciones lineales (de orden superior, inclusive), estos resultados fueron un faro matemático que guiaba hacia la Teoría de Números, hasta que el matemático francés Pierre de Fermat (1601–1665) entró en escena.

Una ecuación diofántica es una expresión matemática que debe ser resuelta sólo en números naturales (Enteros) y solo en ellos, [9, 7, 13].

Un problema fundamental de la Teoría asociada al estudio de las ecuaciones diofánticas, surge de la siguiente pregunta: ¿ Dada una ecuación (o sistema de ecuaciones) diofántica, es posible que existan o no soluciones?.

Nosotros aquí abordaremos algunos problemas relacionados con este tipo de preguntas. En el abordaje utilizamos como herramienta básica la función escalar algorítmica, construída a lo largo de toda la sección §3.

En este contexto, los problemas que aquí serán tratados estan asociados a las así llamadas "Conjeturas de Beal y Fermat", ver [1, 4, 5, 6] y las referencias allí contenidas.

La conjetura de Beal

Sean A, B, C, x, y, z todos números naturales con x, y, $z > 2$. Si

$$A^x + B^y = C^z,$$

entonces A, B y C tienen un factor en común.

Una forma equivalente de escribir la conjetura de Beal's es:

La ecuación

$$A^x + B^y = C^z,$$

no tiene solución en números naturales A, B, C, x, y, z , con x, y, $z > 2$ y A, B , C coprimos.

La conjetura de Fermat

La conjetura de Fermat declara que la ecuación

$$A^n + B^n = C^n,$$

no tiene soluciones A, B, $C \in \mathbb{N}$ con $n \in \mathbb{N}$ y $n > 2$.

Observación 4.1.

(i) Para probar las conjeturas de Beal y Fermat es sufiente considerar exponentes x, y, z y n todos primos.

(ii) La conjetura de Fermat – llamada también por los teóricos de los números como "El último Teorema de Fermat"– ha sido recientemente probada (o re-probada) usando herramientas muy sofisticadas. El matemático que logró tal azaña es el Inglés A. Walis (en el año 1996) demostrando para ello otra conjetura llamada la conjetura de Taniyama–Shimura-Weil.

(iii) Nosotros no pretendemos resolver estas conjeturas, sin embargo damos un pequeño Teorema que involucra a ambas. En la prueba de este Teorema, sólo utilizamos el concepto de función escalar algorítmica. En el futuro inmediato esperamos continuar con el estudio de estos y otros interesantes problemas.

Teorema 4.2. Sean $A \in \mathbb{N}$ par y $B \in \mathbb{N}$, $C \in \mathbb{N}$ impares, tales que $\phi_8(B \times C) \neq 1$. Sean además x, y, z todos primos mayores que dos. Entonces la ecuación Diofántica

$$A^x + B^y = C^z,$$

no tiene solución.

Corolario 4.3. Sean $A \in \mathbb{N}$ par y $B \in \mathbb{N}$, $C \in \mathbb{N}$ impares. Si $\phi_8(B \times C) \neq 1$, entonces la ecuación de Fermat

$$A^p + B^p = C^p,$$

no tiene soluciones, cuando p es un primo mayor que 2.

Teorema 4.4 Sean $A \in \mathbb{N}$ par y $B \in \mathbb{N}$, $C \in \mathbb{N}$ impares, tales que $\phi_7(A \times B \times C) \neq 0$. Entonces la ecuación Diofántica

$$A^3 + B^3 = C^3,$$

no tiene solución.

Antes de dar las demostraciones a estos resultados, serán necesarios un par de Lemas técnicos:

Lema 4.5. Sea $B \in \mathbb{N}$ un número impar, entonces $\phi_8(B^2) = 1$, y en general $\phi_8(B^{2n}) = 1$ para todo $n \in \mathbb{N}^*$.

Demostración: En la demostración del Lema usamos el Teorema 2.8, junto con la definición de la función escalar algorítmica ϕ_b y los Corolarios 2.10–2.11 de la sección §2. En

efecto, si $B \in \mathbb{N}$ es impar, entonces el tiene la forma $B = 2k + 1$, con $k \in \mathbb{N}$. Suponga ahora que $k \in \mathbb{N}$ es par, esto implica que $k = 2t$, para algún $t \in \mathbb{N}$, asi $B^2 = 16t^2 + 8t + 1$. Por tanto, para cada $k \in \mathbb{N}$ par, tenemos

$$\begin{aligned} \phi_8(B^2) &= \phi_8(16t^2 + 8t + 1) \overset{(2,10)}{=} \phi_8(\phi_8(16t^2) + \phi_8(8t) + \phi_8(1)) = \\ \overset{(b)}{=} \phi_8(0 + 0 + \phi_8(1)) &= \phi_8(\phi_8(1)) \overset{(c)}{=} \phi_8(1) \overset{def}{=} 1. \end{aligned}$$

Ahora en cambio, si $k \in \mathbb{N}$ es impar, osea de la forma $k = 2t + 1$, entonces

$$B^2 = 4(2t + 1)^2 + 4(2t + 1) + 1,$$

y de aquí, tenemos

$$B^2 = 16t^2 + 16t + 4 + 8t + 4 + 1 = 16t^2 + 24t + 8 + 1$$

De donde, para todo $k \in \mathbb{N}$ impar, se tiene

$$\begin{aligned} \phi_8(B^2) &= \phi_8(16t^2 + 24t + 8+1) \overset{2,10}{=} \\ &= \phi_8(\phi_8(16t^2) + \phi_8(24t) + \phi_8(8) + \phi_8(1)) = \\ \overset{(b)}{=} \phi_8(0 + 0 + 0 + \phi_8(1)) &= \phi_8(\phi_8(1)) = \phi_8(1) \overset{def}{=} 1. \end{aligned}$$

Esto demuestra la primera parte del Lema 4.5.

Para la segunda parte, utilizamos el Corolario 2.11. En efecto, sea $z = B^2$ con $B \in \mathbb{N}$ un número impar, por lo probado anteriormente, tenemos $\phi_8(z) = 1$, de donde

$$\phi_8(B^{2n}) = \phi_8(z^n) \overset{2,11}{=} \phi_8([\phi_8(z)]^n) = \phi_8(1^n) = \phi_8(1) \overset{def}{=} 1$$

para todo $n \in \mathbb{N}^*$. Esto prueba nuestro Lema.

Corolario 4.6. Sea $B \in \mathbb{N}$ un número impar, entonces $\phi_8(B^3) = \phi_8(B)$, y en general $\phi_8(B^{2n+1}) = \phi_8(B)$ para todo $n \in \mathbb{N}^*$.

Demostración:

La demostración del Corolario sigue del siguiente hecho. Note que

$$\begin{aligned} \phi_8(B^3) &= \phi_8(B^2 \times B) = \phi_8(\phi_8(B^2) \times \phi_8(B)) \\ \overset{\text{lema 4.5}}{=} \phi_8(1 \times \phi_8(B)) &= \phi_8(\phi_8(B)) = \phi_8(B), \end{aligned}$$

y análogamente, tenemos

$$\begin{aligned} \phi_8(B^{2n+1}) &= \phi_8(B^{2n} \times B) = \phi_8(\phi_8(B^{2n}) \times \phi_8(B)) \\ \overset{\text{lema 4.5}}{=} \phi_8(1 \times \phi_8(B)) &= \phi_8(\phi_8(B)) = \phi_8(B). \end{aligned}$$

Lema 4.7. Sea $B \in \mathbb{N}$ un número impar, tal que $B < 8$, entonces $\phi_8(B) = B$.

Demostración: La demostración de este Lema es evidente, y sigue de la definición de la función escalar algorítmica ϕ_b con $b = 8$.

Lema 4.8. Sea $B \in \mathbb{N}$ un número par, entonces $\phi_8(B^3) = 0$, y en general $\phi_8(B^{2n+1}) = 0$ para todo $n \in \mathbb{N}$.

Demostración: Si $B \in \mathbb{N}$ es un número par, entonces $B = 2k$, así $\phi_8(B^3) = \phi_8(8k^3) = \phi_8(8z) = 0$, donde $z = k^3 \in \mathbb{N}$. Análogamente, tenemos

$$\phi_8(B^{2n+1}) = \phi_8(2^{2n+1}k^{2n+1}) = \phi_8(\phi_8(2^{2n+1}) \times \phi_8(k^{2n+1})) = \phi_8(0 \times \phi_8(k^{2n+1})) = 0.$$

Lema 4.9. Sea $B \in \mathbb{N}$, tal que $\phi_7(B) \neq 0$, entonces $\phi_7(B^3) = 1$ ó $\phi_7(B^3) = 6$.

Demostración: La demostración sigue el mismo análisis que el Lema 4.5, y por tanto aquí no la daremos.

Ahora, demostramos el Teorema 4.2, para ello argumentamos por contradicción.

Demostración del Teorema 4.2 Suponga que existan $A_0 \in \mathbb{N}$ par y $B_0 \in \mathbb{N}$, $C_0 \in \mathbb{N}$ impares con $\phi_8(B_0 \times C_0) \neq 1$, tales que

$$A_0^x + B_0^y = C_0^z,$$

donde los exponentes x, y y z, en la expresión, son todos primos mayores que dos. Entonces, por el hecho que ϕ_8 es función, tenemos que

$$\phi_8(A_0^x + B_0^y) = \phi_8(C_0^z),$$

de donde $\phi_8(\phi_8(A_0^x) + \phi_8(B_0^y)) = \phi_8(C_0^z)$. Ahora del hecho que A_0 es par, el Lema 4.8 implica $\phi_8(A_0^x) = 0$. Por tanto, tenemos directamente que:

$$\begin{aligned} \phi_8(C_0^z) &= \phi_8(A_0^x + B_0^y) \\ &= \phi_8(\phi_8(A_0^x) + \phi_8(B_0^y)) = \phi_8(0 + \phi_8(B_0^y)) = \phi_8(B_0^y), \end{aligned} \tag{2}$$

ya que $\phi_8(\phi_8(B_0^y)) = \phi_8(B_0^y)$. Ahora, gracias al Corolario 4.6 y a la imparidad de B_0 y C_0, podemos remplazar en la expresión (2) $\phi_8(B_0^y)$ por $\phi_8(B_0)$ y por el otro lado, reemplazamos $\phi_8(C_0^z)$ por $\phi_8(C_0)$. Por tanto, deducimos de (2) que $\phi_8(B_0) = \phi_8(C_0)$. Ahora, multiplique esta igualdad por C_0, así tenemos

$$C_0 \times \phi_8(B_0) = C_0 \times \phi_8(C_0). \tag{3}$$

Aplicamos una vez más ϕ_8 a la expresión (3), de donde surge

$$\phi_8(C_0 \times \phi_8(B_0)) = \phi_8(C_0 \times \phi_8(C_0)) = \phi_8(\phi_8(C_0) \times \phi_8(C_0)) = \phi_8(C_0^2) = 1.$$

Pero por otro lado,

$$\phi_8(C_0 \times \phi_8(B_0)) = \phi_8(\phi_8(C_0) \times \phi_8(B_0)) = \phi_8(C_0 \times B_0).$$

Es decir,

$$1 = \phi_8(C_0 \times B_0) = \phi_8(B_0 \times C_0) \ .$$

Esto contradice claramente la hipótesis del Teorema 4.2. En consecuencia, podemos decir que la Conjetura de Beal, bajo nuestras hipótesis, es verdadera. □

Observación 4.10:

(i) Note que existen infinitos productos de dos números impares que satisfacen la condición $\phi_8(B_0 \times C_0) \neq 1$, por ejemplo, 3×7, 5×13, etc. Pero también existen infinitos productos de dos números impares que satisfacen $\phi_8(B_0 \times C_0) = 1$, por ejemplo, 3×11, 17×1, etc.

(ii) Las pruebas del Corolario 4.3 y el Teorema 4.4, siguen esencialmente el mismo tipo de abordaje que la demostración anterior, y por tanto aquí no la daremos.

Note eso sí, que el Corolario 4.3 es una consecuencia directa del Teorema 4.2. En efecto, basta hacer $x = y = z = p$.

(iii) En esta parte del trabajo, quisimos motivar el uso de la función escalar algorítmica abordando este tipo de conjeturas. Para otros problemas no lineales, lineales generales, y sistemas de ecuaciones lineales diofánticas, la función escalar algorítmica y la función matricial algorítmica deberían, creemos, ser de utilidad para su estudio. Comprobar aquello, será tema de otra investigación.

5. Equivalencias entre la función escalar algorítmica y la congruencia módulo p de Carl F. Gauss

El matemático Johann Carl Friedrich Gauss (Gauss, 1777–1855), apodado el príncipe de las Matemáticas, realizó valiosas contribuciones en diversos campos de las Matemática y la Física, esto lo convierte con el tiempo, al decir de sus pares, en uno de los más importantes científicos del siglo XIX.

Entre sus múltiples descubrientos se hallan diversos resultados sobre divisibilidad. Un interesante aporte, en esta área de acción, fué definir lo que es el concepto de **Congruencia Módulo** p, el cual está asociado a la división de dos núneros enteros. Específicamente, Gauss da la siguiente

Definición 5.1. (K. F. Gauss).
Sean a y b dos enteros. Diremos que a es **congruente** a b módulo p siempre que $a - b$ sea divisible por p.

Observación 5.2.

(a) Se desprende de la Definición 5.1 que a es **congruente** a b módulo p si y sólo si existe un $k \in \mathbb{Z}$ tal que $a = b + kp$.

(b) Dado un cierto número entero p, el conjunto de los números enteros puede ser dividido en clases de equivalencia en las que los elementos de una misma clase son aquellos que poseen el mismo resto al dividirlos por p. El conjunto de las clases de equivalencias así formadas se representan en Teoría de los Números por $\mathbb{Z}/[p]$, que debe leerse $\mathbb{Z}$ módulo p. Cuando dos enteros pertenecen a una misma clase de equivalencia en $\mathbb{Z}/[p]$ decimos que son congruentes módulo p, esto es lo que en esencia dice la Definición 5.1.

Teorema 5.3 (Equivalencia entre ϕ_b y la congruencia módulo)
Sean $a, p \in \mathbb{N}^*$ con $a \geq p$ y $p \geq 2$, tales que,

$$a = r_0 + r_1 p + r_2 p^2 + \ldots + r_{n-1} p^{n-1} + r_n p^n,$$

con $0 \leq r_i \leq p-1$, para todo $0 \leq i \leq n$, entonces

$$\phi_p(a - r_0) = 0 \text{ sí y sólo si } a \text{ es congruente a } r_0 \text{ módulo } p.$$

Observación 5.4. La demostración del Teorema 5.3 es directa. Note además que en el ambiente de los números naturales, obtenemos la triple equivalencia:

$$\begin{aligned} a \text{ es congruente a } r_0 \text{ módulo } p &\iff \phi_p(a - r_0) = 0 \\ &\iff \text{Existe } q \in \mathbb{N}^*, \text{ tal que } a = r_0 + qp. \end{aligned}$$

En otras palabras, la función escalar algorítmica definida en la sección §2 del trabajo, se encuentra entre la congruencia módulo y el algoritmo de Euclides. Si nos remitimos sólo al conjunto $\mathbb{N}$, es lógico pensar, que de alguna forma, debería ser posible demostrar, a través de ϕ_m, la mayoría de los resultados de la Aritmética asociados al concepto de congruencia módulo p. Este debe ser, por su importancia, un tema para futuras investigaciones. En esta dirección, nuestro grupo ha hecho algunos pequeños avances con el pre print titulado "Entre la congruencia de Gauβ y el algoritmo de la división", ver [8].

6. Apéndice

```
Program Resto(input, output) ;
Var  a, b, q, r:  Integer ;
Begin
Writeln ( 'queremos determinar el resto r = φ_b(a)') ;
Writeln ('cuando el entero a se divide entre el entero positivo b') ;
Write ('a =') ;
Read (a) :
Write ('b =') ;
Read (b) :
If a = 0 then
  Writeln ( Én este caso, a = 0, por lo que q = 0 y r = 0.')
Else
Begin
    r := abs(a) ;
    q := 0 ;
While (r >= b) do
 Begin
    r := r − b ;
    q := q + 1 ;
  End
  if a > 0 then
  Begin
   Writeln ( 'cuando dividimos', a : 0, éntre', b : 0.  '.') ;
   Writeln ( ´el resto r =', r : 0,'.') ;
  End
  Else if r = 0 then
  Begin
   Writeln ( 'cuando dividimos', a : 0, éntre', b : 0.  '.') ;
   Writeln (él resto r = 0.') ;
  End
  Else
  Begin
   Writeln ( 'cuando dividimos', a : 0, éntre', b : 0.  '.') ;
   Writeln (él resto r = φ_b(a) =', b − r : 0,'.') ;
   End: End : End
```

Queremos determinar el resto $r = \phi_b(a)$
cuando el entero a se divide entre el entero positivo b.
$a = 43$
$b = 8$
Cuando dividimos 43 entre 8,
el resto $r = \phi_b(43) = 3$

Referencias

[1] R. D, MAULDIN, A Generalization of Fermat's Last Theorem: The Beal's Conjeture and Prize Problem. Notices of the AMS, 44, N11, p.p, 1436–1437 (1997).

[2] A. MENEZES, P. VAN OORSCHOT, and S. VANSTONE, Handbook of Applied Cryptography, CRC Press, (1996).

[3] C. MERCADO, Historia de las Matemáticas, Ed. Universitaria, Santiago de Chile, (1972).

[4] S. LANG, Old and new conjetured diophantine inequalities, Bull. Amer. Math. Soc, 23, p.p, 37–75 (1990).

[5] A. WALIS, Modular elliptic curves and Fermat's Last Theorem, Ann. Math. 141 , p.p, 443–551 (1995).

[6] S. SINGH, O último Teorema de Fermat, Ed. Record, Rio de Janeiro, (1999).

[7] N. L. BIGGS, Matemática Discreta, Ed. Vicens–Vives, (1994).

[8] L.A. CORTES VEGA, D. E. ROJAS CASTRO, Y. S. SANTIAGO AYALA and S. C. ROJAS ROMERO, Entre la congruencia de Gauss y el Algoritmo de Euclides: Una Observación, Pre-Print, (2007).

[9] B. KOLMAN, R. S. BUSBY y S. C. ROSS, Estructura de matemáticas discretas para la computación, Prentice Hall, (1987).

[10] R. GRAHAM, D. E. KNUTH y O. PATASHNIK, Concrete Mathematics: A fondation for Computer Science, Addison-Wesley, (1994).

[11] D. BURTON, Elementary Number Theory, Ed. Allyn–Bacon, (1980).

[12] A. HODGES, A. TURING: The Enigma of Intelligence, Ed. Unwin–Paperbacks, (1983).

[13] R. P. GRIMALDI, Discrete and Combinatorial Mathematics. An Applied Introduction, Addison-Wesley, (1994).