Miscelánea Matemática **74** (2022) 59-77 SMM

DOI: https://doi.org/10.47234/mm.7407

Funciones booleanas, bases de Gröbner y cuasigrupos

A. Castillo Ramírez
Centro Universitario de Ciencias Exactas e Ingenierías,
Universidad de Guadalajara, Guadalajara, Jalisco, México
alonso.castillor@academicos.udg.mx,

I. Romo Alvarado
Centro Universitario de Ciencias Exactas e Ingenierías,
Universidad de Guadalajara, Guadalajara, Jalisco, México
ismael.romo@alumnos.udg.mx

y

María P. Suárez Fernández
Centro Universitario de Ciencias Exactas e Ingenierías,
Universidad de Guadalajara, Guadalajara, Jalisco, México
delapaz.suarez@academicos.udg.mx

1. Introducción

La teoría de cuasigrupos es una de las ramas más antiguas del álgebra y la combinatoria. De manera concisa, un *cuasigrupo* consiste en un conjunto no vacío Q equipado con una operación binaria (llamada genéricamente *multiplicación*) que permite la *división* de sus elementos (es decir, para cualesquiera $a, b \in Q$, las ecuaciones $ax = b$ y $ya = b$ tienen soluciones únicas). Es fácil ver que todo grupo es un cuasigrupo, pero, de manera general, un cuasigrupo puede no contener elemento identidad, ni inversos de sus elementos, y su operación puede ser no asociativa (no necesariamente se cumple que $a(bc) = (ab)c$ para todos los elementos $a, b, c \in Q$).

La tabla de multiplicar de un cuasigrupo finito Q con n elementos es siempre un *cuadrado latino* de $n \times n$; es decir, es una tabla de $n \times n$ con entradas en Q en la que cada elemento de Q ocurre exactamente una vez en cada fila y columna de la tabla. Recíprocamente, todo cuadrado latino define la operación de un cuasigrupo finito. Los cuadrados latinos fueron estudiados originalmente por el matemático coreano

Palabras clave: cuasigrupo, cuadrado latino, función booleana, base de Gröbner.

Choi Seok-Jeong y por Leonhard Euler, y actualmente tienen numerosas aplicaciones en el diseño de experimentos, la teoría de códigos y la criptografía [5, 6, 9].

En este trabajo presentamos la idea, desarrollada en [7], de resolver sistemas de ecuaciones definidas en cuasigrupos de orden 2^n usando funciones booleanas y bases de Gröbner, dos herramientas de áreas ajenas a la teoría de cuasigrupos. Por un lado, las *funciones booleanas* son funciones de la forma $\{0,1\}^n \to \{0,1\}$, lo cual las hace indispensables en la teoría de la computación dado que proveen un modelo fundamental para las operaciones básicas con dígitos binarios (bits). Por otro lado, las *bases de Gröbner* son tipos particulares de conjuntos generadores de ideales en anillos de polinomios en varias variables, lo cual las hace herramientas fundamentales para resolver sistemas de ecuaciones definidos sobre campos.

Además de ser objetos interesantes desde el punto de vista de las matemáticas puras, los cuasigrupos finitos tienen importantes aplicaciones en el desarrollo de esquemas criptográficos. En estos modelos, los mensajes se codifican mediante la operación de un cuasigrupo, y la decodificación se reduce precisamente al problema de resolver un sistema de ecuaciones sobre el cuasigrupo. Referimos al lector a [1] para conocer más detalles sobre las aplicaciones de cuasigrupos en criptografía.

A continuación describimos la estructura de este trabajo. La sección 2 la dedicamos a hablar del conjunto de funciones booleanas, el cual tiene estructura de anillo conmutativo con identidad, y la llamada *forma normal algebraica* de una función booleana. En la sección 3 presentamos una breve introducción a los cuasigrupos y mostramos cómo describir la operación de un cuasigrupo finito Q de orden 2^n en términos de una n-tupla de funciones booleanas. En la sección 4 ilustramos cómo asociar a cada sistema de ecuaciones definido sobre Q un sistema de ecuaciones equivalente definido sobre el campo finito con dos elementos. Debido a que las ecuaciones del sistema no son necesariamente lineales, en la sección 5 revisamos el concepto de bases de Gröbner, el cual es una herramienta muy poderosa para simplificar el sistema y finalmente encontrar las soluciones. A lo largo de todas las secciones, ilustramos nuestros cálculos usando el paquete Sage [10].

2. Funciones booleanas

En esta sección definimos lo que es una función booleana y mostramos cómo obtener su forma normal algebraica.

Definición 2.1. Sea $n \in \mathbb{N}$. Una **función booleana n-aria** es simplemente una función de la forma

$$f : \{0,1\}^n \to \{0,1\},$$

donde

$$\{0,1\}^n := \{(a_1, a_2, \ldots, a_n) : a_i \in \{0,1\}\}.$$

Dependiendo del contexto, el conjunto $\{0,1\}$ puede identificarse con distintas estructuras, como un campo o un álgebra booleana (véase [4]). Sin embargo, en este trabajo, identificaremos al conjunto $\{0,1\}$ con $\mathbb{F}_2$, el campo con dos elementos (donde la suma y multiplicación se realizan módulo 2).

Denotamos por $\mathcal{B}_n$ al conjunto de funciones booleaneas n-arias. Este conjunto puede verse como un anillo conmutativo con identidad si lo equipamos con las operaciones puntuales inducidas por $\mathbb{F}_2$; es decir, definimos la suma y el producto de funciones booleanas de la siguiente forma: dadas $f : \mathbb{F}_2^n \to \mathbb{F}_2$ y $g : \mathbb{F}_2^n \to \mathbb{F}_2$,

$$(f+g)(x) := f(x) + g(x) \quad \text{y} \quad (fg)(x) := f(x)g(x), \quad \forall x \in \mathbb{F}_2^n.$$

Una forma conveniente de definir a una función booleana es mediante su tabla de verdad.

Definición 2.2. Sea $f : \mathbb{F}_2^n \to \mathbb{F}_2$ una función booleana. La **tabla de verdad** de f es una tabla $V_f := [v_{i,j}]$ de $2^n \times 2$, donde la primera columna de la tabla enlista a todos los elementos de $\mathbb{F}_2^n$ y la entrada $v_{i,2}$ contiene a $f(v_{i,1})$.

Es importante observar que la tabla de verdad V_f depende del orden en el que aparezcan los elementos de $\mathbb{F}_2^n$ en la primera columna. Dos maneras comunes de ordenar a los elementos en un producto cartesiano de un conjunto totalmente ordenado (por ejemplo, $\mathbb{F}_2^n$, donde asumimos que $0 < 1$) son las siguientes:

- **Orden lexicográfico**: $(a_1, a_2, \ldots, a_n) < (b_1, b_2, \ldots, b_n)$ si y solo si $a_i < b_i$, donde i es la coordenada más pequeña donde las tuplas difieren.
- **Orden lexicográfico inverso**: $(a_1, a_2, \ldots, a_n) < (b_1, b_2, \ldots, b_n)$ si y solo si $a_i < b_i$ donde i es la coordenada más grande donde las tuplas difieren.

Ejemplo 2.3. Sea $f \in \mathcal{B}_3$ la función booleana definida por la siguiente tabla de verdad:

(x_1, x_2, x_3)	$f(x_1, x_2, x_3)$
$(0,0,0)$	1
$(1,0,0)$	1
$(0,1,0)$	0
$(1,1,0)$	1
$(0,0,1)$	0
$(1,0,1)$	1
$(0,1,1)$	0
$(1,1,1)$	1

Observemos que la primera columna de la tabla está ordenada de forma creciente por el orden lexicográfico inverso.

Otra manera de representar a una función booleana es mediante su *forma normal algebraica*, pero para justificar esta representación debemos repasar algunos conceptos de teoría de anillos. Sea $R := \mathbb{F}_2[x_1, \ldots, x_n]$ el anillo de polinomios con coeficientes en $\mathbb{F}_2$ en las variables $x_1, x_2, \ldots, x_n$. Recordemos que, dado un ideal I de R, podemos formar al anillo cociente $R/I := \{r + I : r \in R\}$ con las operaciones usuales entre clases laterales:

$$(r + I) + (s + I) = r + s + I \quad \text{y} \quad (r + I)(s + I) = rs + I.$$

Teorema 2.4. *El anillo $\mathcal{B}_n$ de funciones booleanas n-arias es isomorfo al anillo cociente $\mathbb{F}_2[x_1, \ldots, x_n]/I$ donde*

$$I := \left\langle x_1^2 + x_1,\ \ldots,\ x_n^2 + x_n \right\rangle$$

es el ideal generado por todos los polinomios de la forma $x_i^2 + x_i$.

Explícitamente, el isomorfismo de R/I a $\mathcal{B}_n$ manda una clase lateral $f(x_1, \ldots, x_n) + I$ a la función booleana definida por el polinomio $f(x_1, \ldots, x_n)$; la demostración completa del teorema se puede consultar en [2]. Intuitivamente, la razón de identificar a todos los polinomios de la forma $x_i^2 + x_i$ con el elemento cero en el anillo cociente R/I se debe a que $1^2 + 1 = 0$ y $0^2 + 0 = 0$ en el campo $\mathbb{F}_2$.

Para intentar descifrar la forma que tienen los elementos del anillo cociente del teorema anterior, observemos que, para toda $i = 1, 2, \ldots, n$,

$$(x_i^2 + x_i) + I = 0 + I.$$

Esto implica que,

$$x_i^2 + I = x_i + I,$$

y, por inducción, es fácil demostrar que

$$x_i^n + I = x_i + I \quad \forall i = 1, 2, \ldots, n.$$

Con esta identidad, podemos reescribir cualquier clase lateral $p(x)+I$, con $p(x) \in \mathbb{F}_2[x_1, \ldots, x_n]$, como una de la forma

$$\sum_{\alpha \in \mathbb{F}_2^n} c_\alpha x^\alpha + I,$$

donde $c_\alpha \in \mathbb{F}_2$ y

$$\alpha := (\alpha_1, \alpha_2, \ldots, \alpha_n) \in \mathbb{F}_2^n,$$
$$x^\alpha := x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}.$$

Cada polinomio en $\mathbb{F}_2[x_1, \ldots, x_n]$ puede ser visto como una función booleana, la cual resulta al evaluar cada una de las variables en elementos de $\mathbb{F}_2$. Todos los elementos de una clase lateral en $\mathbb{F}_2[x_1, \ldots, x_n]/I$ representan a una misma función booleana, porque los elementos del ideal I son exactamente los polinomios que representan a la función cero. Además, la discusión del párrafo anterior nos permite demostrar que toda $f \in \mathcal{B}_n$ puede ser representada de forma *única* por un polinomio de la forma $\sum_{\alpha \in \mathbb{F}_2^n} c_\alpha x^\alpha$; abusando de la notación, escribimos

$$f(x_1, x_2, \ldots, x_n) = \sum_{\alpha \in \mathbb{F}_2^n} c_\alpha x^\alpha.$$

A esta representación de las funciones booleanas se le conoce como la **forma normal algebraica** (FNA).

Ejemplo 2.5. Veamos algunos ejemplos de cómo obtener la FNA de una función booleana $f : \mathbb{F}_2^n \to \mathbb{F}_2$ definida por su tabla de verdad.

1) Con $n = 1$, consideremos función booleana $f : \mathbb{F}_2 \to \mathbb{F}_2$ definida por la tabla de verdad

(x)	$f(x)$
(0)	1
(1)	0

.

La forma normal algebraica tiene la forma general

$$f(x) = a_0 + a_1 x. \tag{1}$$

Evaluando f en su dominio se tiene que $f(0) = a_0$ y $f(1) = a_0 + a_1$. Lo cual nos conduce al siguiente sistema de ecuaciones lineales

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} f(0) \\ f(1) \end{pmatrix}.$$

Resolviendo el sistema, obtenemos que $a_0 = 1$ y $a_1 = 1$, por lo que la FNA de f es $f(x) = 1 + x$.

2) Con $n = 2$, consideremos la función booleana $f : \mathbb{F}_n^2 \to \mathbb{F}_2$ definida por la tabla de verdad

(x_0, x_1)	$f(x_0, x_1)$
$(0,0)$	1
$(0,1)$	1
$(1,0)$	0
$(1,1)$	1

.

La forma normal algebraica de f tiene la forma

$$f(x) = a_{00} + a_{01}x_1 + a_{10}x_0 + a_{11}x_0x_1. \tag{2}$$

Evaluando f en su dominio se tiene que $f(0,0) = a_{00}$, $f(0,1) = a_{00} + a_{01}$, $f(1,0) = a_{00} + a_{10}$ y $f(1,1) = a_{00} + a_{01} + a_{10} + a_{11}$. Lo cual nos conduce al siguiente sistema de ecuaciones lineales

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}.$$

Resolviendo el sistema obtenemos que $a_{00} = 1$, $a_{01} = 0$, $a_{10} = 1$ y $a_{11} = 1$, por lo que la FNA de f es

$$f(x_0, x_1) = 1 + x_0 + x_0x_1.$$

Para calcular la forma normal algebraica de f usando el paquete Sage [10], es necesario introducir la segunda columna de la tabla de verdad de f pero con los elementos del dominio ordenados de manera creciente respecto al orden lexicográfico inverso. La tabla de verdad de f respecto a este orden es:

(x_0, x_1)	$f(x_0, x_1)$
$(0,0)$	1
$(1,0)$	0
$(0,1)$	1
$(1,1)$	1

A continuación ilustramos cómo obtener la forma normal algebraica de f en Sage:

```
sage: from sage.crypto.boolean_function import BooleanFunction
sage: B1=BooleanFunction([1,0,1,1])
sage: B1.algebraic_normal_form()
x0*x1 + x0 + 1
```

3) Para $n = 3$, consideremos la función booleana $f : \mathbb{F}_2^3 \to \mathbb{F}_2$ dada por la tabla de verdad

(x_0, x_1, x_2)	$f(x_0, x_1, x_2)$
$(0,0,0)$	1
$(0,0,1)$	1
$(0,1,0)$	0
$(0,1,1)$	1
$(1,0,0)$	0
$(1,0,1)$	1
$(1,1,0)$	0
$(1,1,1)$	1

En este caso la forma normal algebraica tiene la forma

$$f(x_0,x_1,x_2) = a_{000} + a_{001}x_2 + a_{010}x_1 + a_{011}x_1x_2 + \\ +a_{100}x_0 + a_{101}x_0x_2 + a_{110}x_0x_1 + a_{111}x_0x_1x_2. \tag{3}$$

Evaluando f en su dominio se tiene que:

$$f(0,0,0) = a_{000}$$

$$f(0,0,1) = a_{000} + a_{001}$$

$$f(0,1,0) = a_{000} + a_{010}$$

$$f(0,1,1) = a_{000} + a_{001} + a_{010} + a_{011}$$

$$f(1,0,0) = a_{000} + a_{100}$$

$$f(1,0,1) = a_{000} + a_{001} + a_{100} + a_{101}$$

$$f(1,1,0) = a_{000} + a_{010} + a_{100} + a_{110}$$

$$f(1,1,1) = a_{000} + a_{001} + a_{010} + a_{011} + a_{100} + a_{101} + a_{110} + a_{111}.$$

Lo cual nos conduce al siguiente sistema de ecuaciones lineales

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_{000} \\ a_{001} \\ a_{010} \\ a_{011} \\ a_{100} \\ a_{101} \\ a_{110} \\ a_{111} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

Resolviendo este sistema obtenemos que $a_{000} = 1, a_{001} = 0, a_{010} = 1, a_{011} = 1, a_{100} = 1, a_{101} = 1, a_{110} = 1$ y $a_{111} = 1$, por lo que la FNA de f es

$$f(x_0,x_1,x_2) = 1 + x_1 + x_1x_2 + x_0 + x_0x_2 + x_0x_1 + x_0x_1x_2.$$

Ahora, para calcular la forma normal algebraica de f en Sage primero reacomodamos la tabla de verdad respecto al orden lexicográfico inverso:

(x_0, x_1, x_2)	$f(x_0, x_1, x_2)$
$(0,0,0)$	1
$(1,0,0)$	0
$(0,1,0)$	0
$(1,1,0)$	0
$(0,0,1)$	1
$(1,0,1)$	1
$(0,1,1)$	1
$(1,1,1)$	1

Luego, las instrucciones para obtener la forma normal algebraica en Sage son las siguientes:

```
sage: from sage.crypto.boolean_function import BooleanFunction
sage: B3=BooleanFunction([1,0,0,0,1,1,1,1])
sage: B3.algebraic_normal_form()
x0*x1*x2 + x0*x1 + x0*x2 + x0 + x1*x2 + x1 + 1
```

En general, la matriz A_n de $2^n \times 2^n$ que se obtiene al plantear el sistema de ecuaciones para obtener los coeficientes de la FNA de una función booleana tiene la siguiente forma:

$$A_n = \begin{pmatrix} A_{n-1} & 0 \\ A_{n-1} & A_{n-1} \end{pmatrix},$$

donde A_{n-1} es la matriz de $2^{n-1} \times 2^{n-1}$ definida inductivamente y

$$A_1 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

El lector entusiasta puede verificar que esto se cumple en los ejemplos previos. Además, es fácil demostrar por inducción que $A_n = A_n^{-1}$, para toda $n \geq 1$. Esto implica que las soluciones del sistema de ecuaciones siempre están dadas por el vector $A_n v$, donde $v \in \mathbb{F}^{2^n}$ es la segunda columna de la tabla de verdad de la función booleana (respecto al orden lexicográfico).

Para leer más sobre funciones booleanas y la forma normal algebraica, referimos al lector a [2, 4].

3. La conexión entre funciones booleanas y cuasigrupos

Esta sección tiene como objetivo describir la conexión entre funciones booleanas y cuasigrupos de orden 2^n, y mostrar cómo podemos usar esto para resolver un sistema de ecuaciones con coeficientes en el cuasigrupo. Esta idea fue presentada originalmente en el artículo [7].

Comenzamos con la definición de cuasigrupo.

Definición 3.1. Sea Q un conjunto no vacío equipado con una operación binaria $* : Q \times Q \to Q$. Decimos que Q es un **cuasigrupo** si para cada $a, b \in Q$ existen únicos $x, y \in Q$ tales que

$$x * a = b \quad \text{y} \quad a * y = b.$$

Claramente, todo grupo G es un cuasigrupo, pues, dados $a, b \in G$ los únicos $x, y \in G$ que cumplen $x * a = b$ y $a * y = b$ son $x = b * a^{-1}$ y $y = a^{-1} * b$. Sin embargo, un cuasigrupo no necesariamente debe tener elemento identidad, ni inversos para todos sus elementos, ni la operación $*$ tiene que ser asociativa.

Ejemplo 3.2. Los números enteros $\mathbb{Z}$ con la resta $-$ forman un cuasigrupo, pues dados $a, b \in \mathbb{Z}$ los únicos $x, y \in \mathbb{Z}$ que cumplen $x - a = b$ y $a - y = b$ son los enteros $x = b + a$ y $y = a - b$. Sin embargo, este cuasigrupo no es un grupo, pues la resta no es asociativa; en general,

$$a - (b - c) \neq (a - b) - c.$$

Observación 3.3. Todo cuasigrupo Q cumple la propiedad cancelativa. Para comprobar esto supongamos que $x_1 * a = x_2 * a$, para algunos $a, x_1, x_2 \in Q$. Si $b := x_1 * a = x_2 * a$, la unicidad de un elemento $x \in Q$ que cumpla $x * a = b$ implica que $x_1 = x_2$. Similarmente verificamos que $a * y_1 = a * y_2$ implica que $y_1 = y_2$.

El *orden* de un cuasigrupo, es simplemente la cardinalidad del conjunto Q. Cuando el cuasigrupo Q es finito, podemos formar su *tabla de Cayley*, la cual contiene a todos los posibles productos de los elementos de Q. La propiedad de cancelación garantiza que en la tabla de Cayley no se repitan elementos por filas ni por columnas; esta propiedad caracteriza el siguiente concepto:

Definición 3.4. Sea Q un conjunto con n elementos. Un **cuadrado latino** es un arreglo de $n \times n$ con entradas en Q en el cual no se repiten elementos por filas ni por columnas.

Los cuadrados latinos tienen aplicaciones en diversas ramas de las matemáticas como estadística, teoría de códigos, criptografía y teoría de gráficas [5, 6, 9]. Para cada $n \in \mathbb{N}$, es fácil demostrar la existencia de al

menos un cuadro latino de $n \times n$. Sin embargo, no es trivial determinar el número total de cuadros latinos de $n \times n$.[1] La importancia de los cuadrados latinos en nuestro contexto es que resulta que un conjunto finito Q con operación binaria $*$ es un cuasigrupo si y solo si su tabla de Cayley es un cuadrado latino.

Si Q es un cuasigrupo cuyo orden es una potencia de 2, digamos 2^n, podemos hacer una *identificación* de Q con $\mathbb{F}_2^n$ mediante una función biyectiva $\varphi : Q \to \mathbb{F}_2^n$. Con esto, para cualquier $x \in Q$ existen $a_1, \ldots, a_n \in \mathbb{F}_2$ tales que

$$\varphi(x) = (a_1, \ldots, a_n).$$

Por lo tanto, la operación binaria $* : Q \times Q \to Q$ queda identificada con una función de la forma $*_\varphi : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n$, donde el conjunto $\mathbb{F}_2^n \times \mathbb{F}_2^n$ es equivalente a $\mathbb{F}_2^{2n}$. Las funciones coordenadas de $*_\varphi$ son funciones booleanas de la forma $f_i : \mathbb{F}_2^{2n} \to \mathbb{F}_2$ y satisfacen que, para toda $u, v \in \mathbb{F}_2^n$,

$$u *_\varphi v = (f_1(u,v),\ f_2(u,v),\ \ldots,\ f_n(u,v)).$$

Ejemplo 3.5. Sea $Q = \{a, b, c, d\}$ el cuasigrupo definido por la siguiente tabla de Cayley

$*$	a	b	c	d
a	a	b	c	d
b	b	c	d	a
c	c	d	a	b
d	d	a	b	c

Consideremos la identificación $\varphi : Q \to \mathbb{F}_2^2$ dada por

$$\begin{aligned} \varphi(a) &= (0,0), \\ \varphi(b) &= (1,0), \\ \varphi(c) &= (0,1), \\ \varphi(d) &= (1,1). \end{aligned}$$

Entonces, la tabla de Cayley correspondiente a la operación $*_\varphi$ es la siguiente

$*_\varphi$	$(0,0)$	$(1,0)$	$(0,1)$	$(1,1)$
$(0,0)$	$(0,0)$	$(1,0)$	$(0,1)$	$(1,1)$
$(1,0)$	$(1,0)$	$(0,1)$	$(1,1)$	$(0,0)$
$(0,1)$	$(0,1)$	$(1,1)$	$(0,0)$	$(1,0)$
$(1,1)$	$(1,1)$	$(0,0)$	$(1,0)$	$(0,1)$

[1] El lector puede consultar en [8] el resultado que determina la cantidad de cuadros latinos de $n \times n$.

Podemos reescribir esta tabla de Cayley como:

$(x_0, x_1) *_\varphi (x_2, x_3)$	$(f_1(x_0, x_1, x_2, x_3), f_2(x_0, x_1, x_2, x_3))$
$(0,0) *_\varphi (0,0)$	$(0,0)$
$(1,0) *_\varphi (0,0)$	$(1,0)$
$(0,1) *_\varphi (0,0)$	$(0,1)$
$(1,1) *_\varphi (0,0)$	$(1,1)$
$(0,0) *_\varphi (1,0)$	$(1,0)$
$(1,0) *_\varphi (1,0)$	$(0,1)$
$(0,1) *_\varphi (1,0)$	$(1,1)$
$(1,1) *_\varphi (1,0)$	$(0,0)$
$(0,0) *_\varphi (0,1)$	$(0,1)$
$(1,0) *_\varphi (0,1)$	$(1,1)$
$(0,1) *_\varphi (0,1)$	$(0,0)$
$(1,1) *_\varphi (0,1)$	$(1,0)$
$(0,0) *_\varphi (1,1)$	$(1,1)$
$(1,0) *_\varphi (1,1)$	$(0,0)$
$(0,1) *_\varphi (1,1)$	$(1,0)$
$(1,1) *_\varphi (1,1)$	$(0,1)$

De esta manera es más sencillo encontrar las dos funciones booleanas $f_1, f_2 \in \mathcal{B}_4$ tales que

$$(x_0, x_1) *_\varphi (x_2, x_3) = (f_1(x_0, x_1, x_2, x_3),\ f_2(x_0, x_1, x_2, x_3)).$$

Usando el método visto en la sección anterior, encontramos que las formas normales algebraicas de f_1 y f_2 son

$$\begin{aligned} f_1(x_0, x_1, x_2, x_3) &= x_0 + x_2, \\ f_2(x_0, x_1, x_2, x_3) &= x_0x_2 + x_1 + x_3. \end{aligned}$$

Por lo tanto, la operación del cuasigrupo está dada por

$$(x_0, x_1) *_\varphi (x_2, x_3) = (x_0 + x_2, x_0x_2 + x_1 + x_3).$$

4. Ecuaciones sobre cuasigrupos

Probablemente el lector está familiarizado con ecuaciones sobre campos como los números racionales o los números reales. A diferencia de estos casos, cuando hablamos de ecuaciones sobre cuasigrupos debemos considerar que la operación puede ser no asociativa y no conmutativa, por lo que se debe respetar el orden en el que aparecen los paréntesis y las variables. Por ejemplo, en un cuasigrupo las siguientes ecuaciones

no necesariamente son equivalentes:

$$(y_1 * y_2) * (y_1 * y_3) = y_2 * y_1$$
$$y_1 * ((y_2 * y_1) * y_3) = y_1 * y_2.$$

El objetivo de esta sección es mostrar cómo transformar ecuaciones sobre un cuasigrupo de orden 2^n a ecuaciones equivalentes sobre $\mathbb{F}_2$, haciendo uso de la representación de la operación del cuasigrupo como una n-tupla de funciones booleanas.

Dado un elemento a en un cuasigrupo Q, definimos las funciones $I_a : Q \to Q$ y $D_a : Q \to Q$ inducidas por la multiplicación izquierda y derecha por a, respectivamente; es decir,

$$I_a(x) = a * x \quad \text{y} \quad D_a(x) = x * a, \quad \forall x \in Q.$$

Luego, podemos reescribir cualquier expresión en el cuasigrupo como una composición de estas funciones. Por ejemplo,

$$y_1 * (y_2 * y_3) = I_{y_1} \circ D_{y_3}(y_2)$$
$$(y_1 * y_2) * y_3 = D_{y_3} \circ D_{y_2}(y_1),$$
$$(y_1 * (y_2 * y_1)) * y_2 = D_{y_2} \circ I_{y_1} \circ D_{y_1}(y_2)$$

La ventaja de usar composiciones de las funciones multiplicación izquierda y derecha recae en que podemos pensar cualquier expresión sobre el cuasigrupo como un proceso lineal; por ejemplo, la expresión $y_1 * (y_2 * y_3)$ significa «multiplica a y_2 por y_3 por la derecha, y luego multiplica el resultado por y_1 por la izquierda».

Usando los métodos de la sección previa, es sencillo encontrar la FNA de las funciones booleanas asociadas con las operaciones I_a y D_a. Una vez hecho esto, la FNA de las funciones booleanas de cualquier expresión sobre el cuasigrupo puede obtenerse haciendo la composición correspondiente de las funciones booleanas de I_a y D_a. Ilustramos esto con el siguiente ejemplo.

Ejemplo 4.1. Sea $Q = \{0, 1, 2, 3\}$ el cuasigrupo de orden 4 definido por la siguiente tabla de Cayley:

$*$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

Consideremos la identificación $\varphi : Q \to \mathbb{F}_2^2$ dada por

$$\varphi(0) = (0, 0), \ \varphi(1) = (1, 0), \ \varphi(2) = (0, 1), \ \varphi(3) = (1, 1).$$

Calculando las funciones booleanas $f_1, f_2 \in \mathcal{B}_4$ que describen la operación del cuasigrupo, obtenemos que

$$\begin{aligned} f_1(s_0, s_1, s_2, s_3) &= s_0 + s_2, \\ f_2(s_0, s_1, s_2, s_3) &= s_0 s_2 + s_1 + s_3. \end{aligned}$$

Por lo tanto, la operación del cuasigrupo está dada por

$$\begin{aligned} (s_0, s_1) *_\varphi (s_2, s_3) &= (f_1(s_0, s_1, s_2, s_3), f_2(s_0, s_1, s_2, s_3)) \\ &= (s_0 + s_2, s_0 s_2 + s_1 + s_3). \end{aligned}$$

Consideremos la siguiente ecuación sobre Q:

$$(y_1 * (y_2 * y_3)) * y_1 = 1.$$

Usando la notación de multiplicación izquierda y derecha obtenemos que

$$(y_1 * (y_2 * y_3)) * y_1 = D_{y_1} \circ I_{y_1} \circ D_{y_3}(y_2).$$

Identificando las variables $y_1, y_2, y_3 \in Q$ con

$$\begin{aligned} \varphi(y_1) &= (x_0, x_1), \\ \varphi(y_2) &= (x_2, x_3), \\ \varphi(y_3) &= (x_4, x_5), \end{aligned}$$

donde $x_i \in \mathbb{F}_2$, obtenemos la FNA de las funciones booleanas asociadas con la expresión $(y_1 * (y_2 * y_3)) * y_1$,

$$\begin{aligned} &(\varphi(y_1) * (\varphi(y_2) * \varphi(y_3))) * \varphi(y_1) \\ &= D_{\varphi(y_1)} \circ I_{\varphi(y_1)} \circ D_{\varphi(y_3)}(\varphi(y_2)) \\ &= D_{\varphi(y_1)} \circ I_{\varphi(y_1)}(x_2 + x_4, x_2 x_4 + x_3 + x_5) \\ &= D_{\varphi(y_1)}(x_0 + x_2 + x_4, x_0(x_2 + x_4) + x_1 + x_2 x_4 + x_3 + x_5) \\ &= (x_2 + x_4, x_0 + x_2 x_4 + x_3 + x_5) \end{aligned}$$

Igualando lo anterior a $\varphi(1) = (1, 0)$, obtenemos el siguiente sistema de ecuaciones sobre $\mathbb{F}_2$, el cual es equivalente a la ecuación dada sobre el cuasigrupo Q:

$$\begin{aligned} x_2 + x_4 &= 1, \\ x_0 + x_2 x_4 + x_3 + x_5 &= 0. \end{aligned}$$

El conjunto de soluciones de este sistema es

$$\{(x_0, x_1, x_2, x_3, x_4, x_5) \in \mathbb{F}_2^6 : x_0 + x_3 + x_5 = 0\}.$$

Usando la biyección φ, podemos encontrar todas las soluciones de la ecuación $(y_1 * (y_2 * y_3)) * y_1 = 1$ sobre Q, pues $(x_0, x_1, \ldots, x_5)$ es solución del sistema sobre $\mathbb{F}_2$ si y solo si $(\varphi^{-1}(x_0, x_1), \varphi^{-1}(x_2, x_3), \varphi^{-1}(x_4, x_5))$

es solución de la ecuación sobre Q. Por lo tanto, la ecuación sobre Q tiene exactamente $2^5 = 32$ soluciones; algunas de ellas son

$$(y_1, y_2, y_3) = (0, 1, 0), \quad (y_1, y_2, y_3) = (0, 2, 2), \quad (y_1, y_2, y_3) = (0, 2, 3).$$

Al programar el ejemplo anterior en Sage, primero definimos las funciones multiplicación izquierda y derecha, y después una función que depende de dos listas: una con las variables y_i y otra lista de ceros y unos, donde cero indica que la multiplicación será por la izquierda y 1 indica la multiplicación será por la derecha. En este caso específico, las entradas de esta última función serían las listas $(y_2, y_3, y_1.y_1)$ y $(1, 0, 1)$ (ya que esta última representa al orden en el que se hacen la multiplicaciones derecha e izquierda).

```
In[1]: sist(f,np.array([y2,y3,y1,y1]),[1,0,1])
Out[1]: array([x2 + x4, x0 + x2*x4 + x3 + x5])
```

En el ejemplo anterior, resultó sencillo encontrar las soluciones del sistema de ecuaciones sobre $\mathbb{F}_2$. Sin embargo esto no siempre será así, pues, al ser sistemas no necesariamente lineales, los métodos clásicos del álgebra lineal no pueden ser aplicados. En la siguiente sección abordaremos una de las herramientas más poderosos para resolver sistemas de ecuaciones no lineales sobre campos: las bases de Gröbner.

5. Bases de Gröbner

A continuación veremos algunos conceptos básicos para definir formalmente a las bases de Gröbner y enunciar algunas de sus propiedades.

Sea K cualquier campo, y consideremos el anillo de polinomios $K[x_1, \ldots, x_n]$. Recordemos que un *monomio* es un polinomio de la forma $cx_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$, donde $c \in K$ y $\alpha_i \geq 0$. Podemos identificar a los exponentes del monomio con una tupla de enteros no negativos $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n) \in \mathbb{Z}_{\geq 0}$, y, abusando de la notación, reescribimos al monomio como cx^α.

Definición 5.1. Un **orden monomial** $\succ$ sobre $K[x_1, \ldots, x_n]$ es una relación sobre $\mathbb{Z}_{\geq 0}^n$, que satisface lo siguiente:

O1) $\succ$ es una relación de orden total en $\mathbb{Z}_{\geq 0}^n$.
O2) Si $\alpha \succ \beta$ y $\gamma \in \mathbb{Z}_{\geq 0}^n$, entonces $\alpha + \gamma \succ \beta + \gamma$.
O3) $\succ$ es un *buen orden*; es decir, todo subconjunto no vacío de $\mathbb{Z}_{\geq 0}^n$ tiene un elemento mínimo.

Dado un orden monomial $\succ$, escribimos $x^\alpha \succ x^\beta$ si $\alpha \succ \beta$.

Observemos que tanto el orden lexicográfico $>_{lex}$ como el orden lexicográfico inverso $>_{invlex}$ en $\mathbb{Z}_{\geq 0}^n$ son órdenes monomiales, pues ambos cumplen las propiedades O1, O2 y O3.

Sea

$$f = \sum_{\alpha \in J} c_\alpha x^\alpha$$

un polinomio en $K[x_1, \ldots, x_n]$ distinto de cero, donde $J \subseteq \mathbb{Z}^n_{\geq 0}$. Asumimos que $c_\alpha \neq 0$, para toda $\alpha \in J$. Respecto a un orden monomial $\succ$, definimos los siguientes conceptos:

i) El **multigrado** de f se define como

$$\text{multideg}\,(f) = \text{máx}(J),$$

donde el máximo se toma con respecto a $\succ$.

ii) El **coeficiente principal** de f se define como

$$\text{LC}\,(f) = c_{\text{multideg}(f)} \in K.$$

iii) El **monomio principal** de f se define como

$$\text{LM}\,(f) = x^{\text{multideg}(f)}.$$

iv) El **término principal** de f se define como

$$\text{LT}\,(f) = \text{LC}\,(f) \cdot \text{LM}\,(f).$$

Ejemplo 5.2. Sea $f = 4x_1x_2^2x_3 + 4x_3^2 - 5x_1^3 + 7x_1^2x_3^2$. Respecto al orden lexicográfico tenemos

$$\begin{aligned} \text{multideg}\,(f) &= (3,0,0) \\ \text{LC}\,(f) &= -5 \\ \text{LM}\,(f) &= x^3 \\ \text{LT}\,(f) &= -5x^3. \end{aligned}$$

Definición 5.3. Sea I un ideal de $K[x_1, x_2, \ldots, x_n]$. Respecto a un orden monomial fijo, decimos que un subconjunto finito $G = \{g_1, \ldots, g_t\}$ es una **base de Gröbner** de I si

$$\langle \text{LT}\,(g_1), \ldots, \text{LT}\,(g_t) \rangle = \langle \text{LT}\,(I) \rangle,$$

donde

$$\text{LT}(I) := \{cx^\alpha : \exists f \in I \text{ tal que } \text{LT}(f) = cx^\alpha\}.$$

Las bases de Gröbner tienen las siguientes propiedades (las demostraciones pueden consultarse en [3]):

1. Todo ideal de $K[x_1, \ldots, x_n]$ distinto de cero tiene una base de Gröbner.
2. Si G es una base de Gröbner del ideal I, entonces $I = \langle G \rangle$.
3. Para todo $f \in I$ existe $g \in G$ tal que $\text{LT}\,(g)$ divide a $\text{LT}\,(f)$.

La **variedad afín** definida por los polinomios $f_1, f_2, \ldots, f_s \in K[x_1, \ldots, x_n]$ es el conjunto

$$\mathbf{V}(f_1, f_2, \ldots, f_s) := \{(a_1, \ldots, a_n) \in K^n : f_i(a_1, \ldots, a_n) = 0, \forall i = 1, \ldots, s\}.$$

En otras palabras, la variedad afín es el conjunto de soluciones del sistema de ecuaciones $f_1(x_1, \ldots, x_n) = \cdots = f_s(x_1, \ldots, x_n) = 0$.

Si $G = \{g_1, \ldots, g_t\}$ es una base de Gröbner de un ideal $I = \langle f_1, \ldots, f_s \rangle$, una propiedad importante es que

$$\mathbf{V}(f_1, \ldots, f_s) = \mathbf{V}(g_1, \ldots, g_t).$$

Es decir, los sistemas de ecuaciones $f_1(x_1, \ldots, x_n) = \cdots = f_s(x_1, \ldots, x_n) = 0$ y $g_1(x_1, \ldots, x_n) = \cdots = g_t(x_1, \ldots, x_n) = 0$ tienen exactamente las mismas soluciones. Debido a que los términos principales de los polinomios f_i son divisibles por los términos principales de los polinomios g_i, las ecuaciones en el segundo sistema serán, en general, más sencillas de resolver.

El método general para encontrar una base de Gröbner de un ideal respecto a un orden monomial es el *algoritmo de Buchberger*, el cual puede pensarse como una generalización del algoritmo de Euclides para polinomios y el método de eliminación gaussiana (véase [3, §2.7]). En Sage, una base de Gröbner G de un ideal I puede calcularse con el siguiente comando:

```
G = I.groebner_basis()
```

En el siguiente ejemplo mostramos cómo resolver un sistema de ecuaciones sobre un cuasigrupo de orden 2^n con ayuda de funciones booleanas y bases de Gröbner.

Ejemplo 5.4. Sea $Q = \{0, 1, 2, 3\}$ el cuasigrupo definido por la siguiente tabla de Cayley:

$*$	0	1	2	3
0	3	2	1	0
1	0	1	2	3
2	2	3	0	1
3	1	0	3	2

Consideremos la identificación $\varphi : Q \to \mathbb{F}_2^2$ definida por

$$\varphi(0) = (0,0),\ \varphi(1) = (1,0),\ \varphi(2) = (0,1),\ \varphi(3) = (1,1).$$

Las funciones booleanas que representan la operación del cuasigrupo son

$$f_1(x_0, x_1, x_2, x_3) = x_0 + x_1 + x_2 + 1,$$
$$f_2(x_0, x_1, x_2, x_3) = x_0 + x_3 + 1.$$

Por lo tanto, la operación del cuasigrupo está representada por

$$(x_0, x_1) *_\varphi (x_2, x_3) = (x_0 + x_1 + x_2 + 1, x_0 + x_3 + 1).$$

Supongamos que queremos resolver el siguiente sistema de ecuaciones:

$$\begin{aligned}(y_1 * (y_2 * y_3)) * y_1 &= 0\\ (y_3 * y_1) * y_2 &= 3\end{aligned}$$

Consideramos la identificación de las variables y_1, y_2 y y_3,

$$\begin{aligned}\varphi(y_1) &= (x_0, x_1),\\ \varphi(y_2) &= (x_2, x_3),\\ \varphi(y_3) &= (x_4, x_5),\end{aligned}$$

donde $x_i \in \mathbb{F}_2$. Usando el método presentado en la sección anterior, obtenemos que el sistema de ecuaciones sobre Q es equivalente al siguiente sistema de ecuaciones sobre $\mathbb{F}_2$:

$$\begin{aligned}x_0 + x_1 + x_3 + x_4 + x_5 + 1 &= 0\\ x_0 + x_2 + x_3 + x_4 + 1 &= 0\\ x_0 + x_1 + x_2 + x_5 &= 0\\ x_0 + x_3 + x_4 + x_5 + 1 &= 0.\end{aligned}$$

La base de Gröbner del ideal definido por los polinomios de este sistema es $G = \{x_0, x_1, x_2+x_5, x_3+x_4+x_5+1\}$; por lo tanto, el siguiente sistema tiene las mismas soluciones que el anterior:

$$\begin{aligned}x_0 &= 0\\ x_1 &= 0\\ x_2 + x_5 &= 0\\ x_3 + x_4 + x_5 + 1 &= 0.\end{aligned}$$

Es sencillo probar que el conjunto de soluciones de este sistema es

$$\{(0,0,0,1,0,0), (0,0,1,0,0,1), (0,0,0,0,1,0), (0,0,1,1,1,1)\}.$$

Por lo tanto, las soluciones del sistema original sobre Q son

$$\{(0,2,0), (0,1,2), (0,0,1), (0,3,3)\}.$$

En el ejemplo anterior, el sistema de ecuaciones obtenido sobre $\mathbb{F}_2$ resultó ser lineal, por lo que también sería posible resolverlo con los métodos del álgebra lineal. En el siguiente ejemplo, el sistema de ecuaciones sobre $\mathbb{F}_2$ es no lineal, por lo que el uso de la técnica de bases de Gröbner es indispensable.

Ejemplo 5.5. Sea $Q = \{0, 1, 2, 3\}$ el cuasigrupo definido por la siguiente tabla de Cayley:

$*$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2.

Consideremos la identificación $\varphi : Q \to \mathbb{F}_2^2$ definida por

$$\varphi(0) = (0,0),\ \varphi(1) = (1,0),\ \varphi(2) = (0,1),\ \varphi(3) = (1,1).$$

Las funciones booleanas que representan al cuasigrupo son

$$\begin{aligned} f_1(x_0, x_1, x_2, x_3) =& x_0 + x_2, \\ f_2(x_0, x_1, x_2, x_3) =& x_0x_2 + x_1 + x_3. \end{aligned}$$

Nuevamente, supongamos que queremos resolver el siguiente sistema de ecuaciones:

$$\begin{aligned} (y_1 * (y_2 * y_3)) * y_1 &= 0 \\ (y_3 * y_1) * y_2 &= 3 \end{aligned}$$

Usando el método presentado en la sección anterior, obtenemos que el sistema de ecuaciones sobre Q es equivalente al siguiente sistema de ecuaciones sobre $\mathbb{F}_2$:

$$\begin{aligned} x_2 + x_4 &= 0 \\ x_0 + x_2x_4 + x_3 + x_5 &= 0 \\ x_0 + x_2 + x_4 + 1 &= 0 \\ x_0x_2 + x_0x_4 + x_1 + x_2x_4 + x_3 + x_5 + 1 &= 0 \end{aligned}$$

A diferencia del ejemplo anterior, este sistema de ecuaciones es no lineal; sin embargo, al encontrar la bases de Gröbner para el ideal correspondiente, obtenemos el siguiente sistema de ecuaciones que tiene las mismas soluciones:

$$\begin{aligned} x_0 + 1 &= 0 \\ x_1 &= 0 \\ x_2 + x_4 &= 0 \\ x_3 + x_4 + x_5 + 1 &= 0 \end{aligned}$$

Con esto, encontramos las soluciones del sistema de ecuaciones sobre $\mathbb{F}_2$:

$$\{(1,0,0,1,0,0), (1,0,0,0,0,1), (1,0,1,0,1,0), (1,0,1,1,1,1)\}.$$

Por lo tanto, las soluciones del sistema de ecuaciones original sobre Q son

$$\{(1,2,0),(1,0,2),(1,1,1),(1,3,3)\}.$$

6. Conclusiones

En este artículo presentamos el método propuesto en [7] para resolver un sistema de ecuaciones sobre un cuasigrupo Q de orden 2^n. A continuación resumimos el procedimiento:

1. Definir una identificación $\varphi : Q \to \mathbb{F}_2^n$.
2. Encontrar la FNA de las n funciones booleanas que representan la operación de Q respecto a φ (Secciones 2 y 3).
3. Convertir el sistema de ecuaciones sobre Q a un sistema de ecuaciones sobre $\mathbb{F}_2$ haciendo la composición de las funciones booleanas que representan la multiplicación derecha e izquierda en Q (Sección 4).
4. Resolver el sistema de ecuaciones sobre $\mathbb{F}$. Si este es lineal, pueden usarse métodos del álgebra lineal; en caso contrario, simplificar el sistema usando bases de Gröbner (Sección 5).

Bibliografía

[1] V. A. Artamonov, «Applications of quasigroups to cryptography», *Sarajevo Journal of Mathematics*, vol. 14, núm. 27, 2018, 191–205.

[2] H. Chimal y J. Díaz, «La formal normal algebraica de una función booleana», *Miscelánea Matemática*, núm. 48, 2009, 47–57.

[3] D. Cox, J. Little y D. O'Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, Springer Cham, 2015.

[4] Y. Crama y P. Hammer, *Boolean Functios: Theory, Algorithms, and Applications*, Cambridge University Press, New York, 2011.

[5] J. Dénes y A. D. Keedwell, *Latin squares and their applications*, Academic Press, New York-London, 1974.

[6] J. H. Dénes y A. D. Keedwell, *Latin squares: New developments in the theory and applications*, vol. 46, Academic Press, Amsterdam, 1991, Annals of Discrete Mathematics, Paul Erdös (foreword).

[7] D. Gligoroski, V. Dimitrova y S. Markovski, «Quasigroups as Boolean Functions, Their Equation Systems and Gröbner Bases», en *Gröbner Bases, Coding, and Cryptography*, Springer, Heidelberg, 2007, 415–420.

[8] V. Shcherbacov, «Elements of Quasigroup Theory and Applications», 2017, https://doi.org/10.1201/9781315120058.

[9] A. P. Street y D. J. Street, *Combinatorics of Experimental Design*, Oxford University Press, Inc., 1987.

[10] The Sage Developers, W. Stein, D. Joyner, D. Kohel, J. Cremona y B. Eröcal, «Mathematics software (version 8.9)», 2019, http://www.sagemath.org.