Miscelánea Matemática **48** (2009) 47–57 SMM

La forma normal algebraica de una función booleana

Henry Chimal Dzul, Javier Díaz Vargas

Facultad de Matemáticas, Universidad Autónoma de Yucatán,

México.

henrychimal@gmail.com, jdvargas@uady.mx

Resumen

En este trabajo, usando el algoritmo de la división para polinomios en $K[x_1, \ldots, x_n]$, damos una demostración original de que el anillo $\mathfrak{B}_n$ de funciones booleanas es isomorfo al anillo cociente $\mathbb{F}_2[x_1, \ldots, x_n]/I$, donde $I = \langle x_1^2 \oplus x_1, \ldots, x_n^2 \oplus x_n \rangle$. Este hecho implica la existencia y unicidad (módulo el ideal I) de la forma normal algebraica de una función booleana.

1. Introducción

Una función booleana es una función del espacio vectorial $\mathbb{F}_2^n$ al campo finito $\mathbb{F}_2$. Existen varias maneras de representar a cada una de estas funciones: la tabla de verdad, la forma normal algebraica, el espectro de Fourier, la forma normal numérica, etc. Ésta última fue introducida por C. Carlet y P. Guillot en 1999 (ver [1]). Cada una de estas representaciones tiene sus ventajas y, desde luego, sus desventajas. Una de las principales cualidades de la forma normal algebraica es que permite definir el grado algebraico de una función booleana. A partir de esta definición es posible definir de manera simple dos clases importantes de códigos algebraicos: los códigos lineales de Reed-Muller y los códigos de Kerdock (ver [3]). En criptografía, se requiere que las funciones criptográficas tengan grado algebraico grande, por lo que el grado algebraico de una función booleana juega un papel muy importante en los criptosistemas que usan funciones booleanas.

En la sección 3 de este trabajo demostramos que toda función booleana admite una forma normal algebraica. Para demostrar que esta representación es única, empleamos el algoritmo de la división en varias variables, el cual es introducido en la sección 2.

2. Algoritmo de la división en $K[x_1, \ldots, x_n]$

Si analizamos con cuidado el algoritmo de la división en $K[x]$, notamos que nuestro primer paso es escribir los términos de los polinomios en orden decreciente con respecto al grado de x. El éxito de este algoritmo depende del trabajo sistemático sobre los términos principales de los polinomios involucrados. Así, un orden sobre los monomios en $K[x]$ es esencial. En esta sección estudiamos el algoritmo de la división para polinomios en n variables, cuyo objetivo es dividir al polinomio $f \in K[x_1, \ldots, x_n]$ entre una s-ada ordenada $F = (f_1, f_2, \ldots, f_s)$ de polinomios $f_1, f_2, \ldots, f_s \in K[x_1, \ldots, x_n]$. Todo lo que sigue está basado en la referencia [2], donde se pueden consultar todas las pruebas de los resultados mencionados en esta sección.

Iniciamos nuestro trabajo definiendo un orden sobre los monomios en $K[x_1, \ldots, x_n]$. Observemos que la asignación $x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \mapsto \alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}_{\geq 0}^n$ establece una correspondencia biyectiva entre los monomios en $K[x_1, \ldots, x_n]$ y $\mathbb{Z}_{\geq 0}^n$. Por lo tanto, cualquier orden $>$ que definamos sobre el espacio $\mathbb{Z}_{\geq 0}^n$ dará lugar a un orden sobre los monomios en $K[x_1, \ldots, x_n]$: si $\alpha > \beta$, de acuerdo a este orden, diremos que $x^\alpha > x^\beta$. Existen varios órdenes sobre $\mathbb{Z}_{\geq 0}^n$ pero para nuestros propósitos muchos de ellos no nos serán útiles, ya que requeriremos órdenes que sean compatibles con la estructura algebraica del anillo $K[x_1, \ldots, x_n]$. Con esta consideración, tenemos la siguiente definición.

Definición 1. Un *orden monomial* sobre $\mathbb{Z}_{\geq 0}^n$ es cualquier relación $>$ que satisface:

1. $>$ es un orden total (o lineal) sobre $\mathbb{Z}_{\geq 0}^n$.

2. Si $\alpha > \beta$ y $\gamma \in \mathbb{Z}_{\geq 0}^n$, entonces $\alpha + \gamma > \beta + \gamma$.

3. $>$ es un buen orden sobre $\mathbb{Z}_{\geq 0}^n$.

La condición (3), de buen orden, significa que cada subconjunto no vacío de $\mathbb{Z}_{\geq 0}^n$ tiene un elemento menor con respecto a $>$ y es equivalente a la siguiente afirmación: para cada cadena infinita estrictamente decreciente $\alpha_1 > \alpha_2 > \alpha_3 > \cdots$ en $\mathbb{Z}_{\geq 0}^n$, existe un entero positivo k tal que $\alpha_k = \alpha_{k+1} = \cdots$.

El orden usual en $\mathbb{Z}_{\geq 0}$, $\cdots > m+1 > m > \cdots > 1 > 0$, satisface las tres condiciones de la definición anterior. Por lo tanto, el orden dado con respecto al grado para monomios en $K[x]$ es un orden monomial. Un orden sobre $\mathbb{Z}_{\geq 0}^n$ (con $n > 1$) es dado a continuación.

Definición 2 (*Orden lexicográfico*)**.** Sean $\alpha, \beta \in \mathbb{Z}^n_{\geq 0}$. Diremos que $\alpha >_{\text{lex}} \beta$ si, en el vector diferencia $\alpha - \beta \in \mathbb{Z}^n$, la primera entrada distinta de cero más a la izquierda es positiva. Escribiremos $x^\alpha >_{\text{lex}} x^\beta$ si $\alpha >_{\text{lex}} \beta$.

Por ejemplo, en $\mathbb{Z}^3_{\geq 0}$, $\alpha = (1, 2, 0) >_{\text{lex}} \beta = (0, 3, 4)$ ya que $\alpha - \beta = (1, -1, -4)$.

Proposición 1. *El orden lexicográfico* $>_{\text{lex}}$ *es un orden monomial sobre* $\mathbb{Z}^n_{\geq 0}$. □

Es importante notar que existen $n!$ órdenes lexicográficos sobre $\mathbb{Z}^n_{\geq 0}$, que corresponden a cómo los vectores $e_1 = (1, 0, \ldots, 0)$, $e_2 = (0, 1, 0, \ldots, 0)$, …, $e_n = (0, \ldots, 0, 1)$ son asignados a las variables x_1, x_2, …, x_n. Por ejemplo, las variables x_1, x_2, …, x_n pueden ser ordenadas en forma usual asignando $e_i \mapsto x_i$; pero cualquier permutación $\sigma \in S_n$ da lugar a un orden lexicográfico: $e_i \mapsto x_{\sigma(i)}$, $i = 1, \ldots, n$.

Existen otros órdenes monomiales sobre $\mathbb{Z}^n_{\geq 0}$ tales como el *orden graduado lexicográfico* y el *orden graduado lexicográfico inverso* (para más detalles el lector puede consultar [2]).

El siguiente lema es un ejercicio en [2] y será de gran utilidad en la demostración del teorema principal de este trabajo.

Lema 1. *Sea* $>$ *cualquier orden monomial. Entonces* $\alpha \geq 0$ *para todo* $\alpha \in \mathbb{Z}^n_{\geq 0}$. *Además, si* x^α *divide a* x^β, *se tiene que* $x^\alpha \leq x^\beta$.

Demostración. Si $0 > \alpha$ para algún $\alpha \in \mathbb{Z}^n_{\geq 0}$, entonces $0 > \alpha > 2\alpha > 3\alpha > \cdots$ es una cadena infinita estrictamente decreciente en $\mathbb{Z}^n_{\geq 0}$ para la cual no existe k tal que $k\alpha = (k+1)\alpha = \cdots$; lo cual contradice que $>$ es un buen orden. Además, si x^α divide a x^β, entonces $\beta = \alpha + \gamma$ para algún $\gamma \in \mathbb{Z}^n_{\geq 0}$. Ya que $\gamma \geq 0$, tenemos que $\beta = \alpha + \gamma \geq \alpha + 0 = \alpha$. □

Como una consecuencia del lema anterior tenemos que $x_i^2 > x_i$ para cualquier orden monomial $>$ sobre $\mathbb{Z}^n_{\geq 0}$, ya que, en $K[x_1, \ldots, x_n]$, x_i siempre divide a x_i^2.

Definición 3. Sea $f = \sum_\alpha a_\alpha x^\alpha$ un polinomio distintos de cero en $K[x_1, \ldots, x_n]$ y sea $>$ un orden monomial.

1. El *exponente principal* de f es $\text{EP}(f) = \text{máx}\{\alpha \in \mathbb{Z}^n_{\geq 0} : a_\alpha \neq 0\}$.
2. El *coeficiente principal* de f es $\text{CP}(f) = a_{\text{EP}(f)} \in K$.
3. El *monomio principal* de f es $\text{MP}(f) = x^{\text{EP}(f)}$.
4. El *término principal* de f es $\text{TP}(f) = \text{CP}(f) \cdot \text{MP}(f)$.

Ejemplo 1. Sea $f = 4xy^2z + 4z^2 - 5x^3 + 7x^2z^2 \in K[x, y, z]$ y $>_{\text{lex}}$ el orden lexicográfico con $x >_{\text{lex}} y >_{\text{lex}} z$. Entonces $\text{EP}(f) = (3, 0, 0)$, $\text{CP}(f) = -5$, $\text{MP}(f) = x^3$ y $\text{TP}(f) = -5x^3$.

El exponente principal tiene las siguientes propiedades útiles, similares a las que tiene el grado de un polinomio de una sola variable.

Lema 2. *Sean $f, g \in K[x_1, \ldots, x_n]$ polinomios no cero. Entonces* $\text{EP}(f\ g) = \text{EP}(f) + \text{EP}(g)$. *Además, si $f + g \neq 0$,* $\text{EP}(f + g) \leq \text{máx}\{\text{EP}(f), \text{EP}(g)\}$ *y la igualdad se cumple si* $\text{EP}(f) \neq \text{EP}(g)$.

Demostración. La igualdad se sigue de la parte 2 de la definición 1 de orden monomial y de que la multiplicación en $K[x_1, \ldots, x_n]$ es distributiva con respecto a la adición. Para demostrar la desigualdad, basta observar que al ordenar los términos en $f + g$, el término que queda como principal es $\text{TP}(f)$ ó $\text{TP}(g)$. Ya que éstos se pueden cancelar, la desigualdad ocurre. □

Ahora estamos listos para formular el algoritmo de la división en $K[x_1, \ldots, x_n]$, el cual extiende al algoritmo para $K[x]$. El objetivo de éste es dividir al polinomio $f \in K[x_1, \ldots, x_n]$ entre una s-ada ordenada $F = (f_1, f_2, \ldots, f_s)$ de polinomios $f_1, f_2, \ldots, f_s \in K[x_1, \ldots, x_n]$; esto significa expresar a f en la forma

$$f = a_1f_1 + a_2f_2 + \cdots + a_sf_s + r,$$

donde los *coeficientes* $a_1, a_2, \ldots, a_s$ y el residuo r están en $K[x_1, \ldots, x_n]$. El lema 2, el cual es propuesto como un ejercicio en [2], es parte importante de la demostración de este teorema.

Teorema 1 (Algoritmo de la división en $K[x_1, \ldots, x_n]$)**.** *Sea $>$ un orden monomial fijo sobre $\mathbb{Z}^n_{\geq 0}$ y $F = (f_1, f_2, \ldots, f_s)$ una s-ada ordenada de polinomios en $K[x_1, \ldots, x_n]$. Entonces cada $f \in K[x_1, \ldots, x_n]$ puede ser escrito como*

$$f = a_1f_1 + a_2f_2 + \cdots + a_sf_s + r,$$

donde $a_i, r \in K[x_1, \ldots, x_n]$ y $r = 0$ ó r es una combinación lineal, con coeficientes en K, de monomios, ninguno de los cuales es divisible por cualquier $\text{TP}(f_1), \text{TP}(f_2), \ldots, \text{TP}(f_s)$. *Llamaremos a r el* **`residuo`** *de la división de f por F. Además, si $r \neq 0$,* $\text{EP}(f) \geq \text{EP}(r)$ *y si $a_if_i \neq 0$, entonces tenemos que* $\text{EP}(f) \geq \text{EP}(a_if_i)$. □

Para ver cómo funciona el algoritmo de la división en $K[x_1, \ldots, x_n]$, considere el ejemplo siguiente.

Ejemplo 2. Dividiremos $f = -x^2y - x^2z + x^3 + x + xyz$ entre $F = (f_1, f_2)$, donde $f_1 = x^2y - z$ y $f_2 = -1 + xy$. Usaremos el orden $>_{\text{lex}}$ con $x >_{\text{lex}} y >_{\text{lex}} z$. El primer paso es escribir los términos de los polinomios en orden decreciente con respecto a $>_{\text{lex}}$:

$$f = x^3 - x^2y - x^2z + xyz + x, \quad f_1 = x^2y - z, \quad f_2 = xy - 1.$$

Coloquemos los divisores f_1 y f_2; los cocientes a_1, a_2 y el residuo r en el siguiente esquema:

$$\begin{array}{rll} a_1: & & \\ a_2: & & r \\ x^2y - z & /x^3 - x^2y - x^2z + xyz + x & \\ xy - 1 & & \end{array}$$

Los términos principales $\text{TP}(f_1) = x^2y$ y $\text{TP}(f_2) = xy$ no dividen al término principal $\text{TP}(f) = x^3$; sin embargo, a diferencia de lo que ocurre en el algoritmo de la división en $K[x]$, $x^3 - x^2y - x^2z + xyz + x$ no es el residuo ya que $\text{TP}(f_1) = x^2y$ y $\text{TP}(f_2) = xy$ dividen a $-x^2y$. Así que, moviendo x^3 al residuo, podemos continuar dividiendo.

$$\begin{array}{rlll} a_1: & & & \\ a_2: & & & r \\ x^2y - z & \underline{/x^3 - x^2y - x^2z + xyz + x} & & \\ xy - 1 & \quad -x^2y - x^2z + xyz + x & \to & x^3 \end{array}$$

Ahora, ambos términos, $\text{TP}(f_1)$ y $\text{TP}(f_2)$, dividen a $\text{TP}(-x^2y - x^2z + xyz + x) = -x^2y$, pero ya que f_1 está listado primero, usamos f_1. Procedemos a dividir $-x^2y$ entre x^2y, obteniendo $a_1 = -1$, y a restar $(-1)f_1$ de $-x^2y - x^2z + xyz + x$.

$$\begin{array}{rlll} a_1: & -1 & & \\ a_2: & & & r \\ x^2y - z & \underline{/x^3 - x^2y - x^2z + xyz + x} & & \\ xy - 1 & \quad -x^2y - x^2z + xyz + x & \to & x^3 \\ & \quad \underline{-x^2y + z} & & \\ & \quad -x^2z + xyz + x - z & & \end{array}$$

Continuando con la división vemos que $\text{TP}(f_1) = x^2y$ y $\text{TP}(f_2) = xy$ no dividen a $\text{TP}(-x^2z + xyz + x - z) = -x^2z$; así que movemos $-x^2z$ al residuo y continuamos dividiendo. El término $\text{TP}(f_1) = x^2y$ no divide

a $\mathrm{TP}(xyz + x + z) = xyz$ pero $\mathrm{TP}(f_2) = xy$ sí. El resultado de esta división es $a_2 = z$ y por lo tanto restamos zf_2 de $xyz + x + z$.

$$\begin{array}{rrcl}
a_1: & -1 & & \\
a_2: & z & & r \\
x^2y - z & /x^3 - x^2y - x^2z + xyz + x & & \\
xy - 1 & \quad -x^2y - x^2z + xyz + x & \to & x^3 \\
& \quad\ -x^2y + z & & \\
\hline
& \quad\ -x^2z + xyz + x - z & & \\
\hline
& \quad\ xyz + x - z & \to & x^3 - x^2z \\
& \quad\ xyz - z & & \\
\hline
& \quad\ x & & \\
\hline
& \quad\ 0 & \to & x^3 - x^2z + x
\end{array}$$

Claramente ninguno de los términos principales de f_1 y f_2 dividen a x; por ello lo movemos al residuo. Por lo tanto

$$f = (-1)(x^2y - z) + z(xy - 1) + (x^3 - x^2z + x), \tag{1}$$

como el lector puede comprobar fácilmente efectuando las operaciones. Si dividimos f entre $G = (f_2, f_1)$ obtenemos que

$$f = (-x + z)(xy - 1) + 0(x^2y - z) + (x^3 - x^2z - z).$$

Comparando esta ecuación con (1) vemos que los a_i y el residuo son diferentes. Esto muestra que ellos pueden cambiar con sólo reordenar los f_i.

3. La Forma Normal Algebraica (F.N.A.)

El conjunto de las funciones booleanas es de gran importancia debido a que, entre otras cosas, las funciones booleanas juegan un papel muy importante en teoría de códigos y criptografía. En esta sección damos una demostración del hecho de que toda función booleana admite una representación polinomial, conocida como la forma normal algebraica.

Sea $\mathbb{F}_2$ el campo finito con dos elementos 0 y 1, con las operaciones de suma $\oplus$ y producto $\otimes$ módulo 2.

Definición 4. Sea $n \in \mathbb{N}$. Una *función booleana* es una función definida sobre el conjunto $\mathbb{F}_2^n$, de todos los vectores binarios de longitud n, al campo $\mathbb{F}_2$. Denotaremos por $\mathfrak{B}_n$ al conjunto de todas las funciones booleanas con dominio $\mathbb{F}_2^n$.

La suma y producto entre funciones booleanas son definidas de manera usual, donde las operaciones son realizadas módulo 2. El conjunto $\mathfrak{B}_n$ (con la suma) es un $\mathbb{F}_2$-espacio vectorial y $|\mathfrak{B}_n| = 2^{2^n}$. Más aún, $\mathfrak{B}_n$ es un anillo conmutativo con uno.

Definimos el *peso* de $f \in \mathfrak{B}_n$ como $w(f) = |\{x \in \mathbb{F}_2^n : f(x) = 1\}|$ y la *distancia de Hamming* en $\mathfrak{B}_n$ es definida por $d(f, g) = w(f \oplus g)$.

Describiremos ahora dos representaciones de las funciones booleanas que, en general, son usadas en criptografía y teoría de códigos.

La *tabla de verdad* o *vector característico* de $f \in \mathfrak{B}_n$ es

$$V_f = (f(v_0), f(v_1), \ldots, f(v_{2^n-1})) \in \mathbb{F}_2^{2^n}, \text{donde } \mathbb{F}_2^n = \{v_0, v_1, \ldots, v_{2^n-1}\}.$$

Note que V_f depende de la biyección $i \mapsto v_i$ mientras que el peso de f no. Las ventajas de esta representación son la simplicidad además de que:

- Establece un isomorfismo entre los $\mathbb{F}_2$-espacios vectoriales $\mathfrak{B}_n$ y $\mathbb{F}_2^{2^n}$. De aquí que la dimensión de $\mathfrak{B}_n$ es 2^n; y
- el peso de f es calculado directamente de su tabla de verdad como $w(f) = \sum_{a \in \mathbb{F}_2^n} f(a)$, donde el símbolo $\sum$ denota a la suma ordinaria de enteros.

Ejemplo 3. Para $n = 2$, consideremos las siguientes funciones booleanas.

(x_1, x_2)	$f(x_1, x_2)$	$g(x_1, x_2)$	$(f \oplus g)(x_1, x_2)$	$(f \otimes g)(x_1, x_2)$
$v_0 = (0, 0)$	1	0	1	0
$v_1 = (0, 1)$	1	0	1	0
$v_2 = (1, 0)$	1	0	1	0
$v_3 = (1, 1)$	0	1	1	0

Notemos que f y g ambas son diferentes de las funciones cero, sin embargo, $f \otimes g$ es la función cero, de lo cual podemos concluir que $\mathfrak{B}_n$ no es un dominio entero. De la tabla podemos ver que $V_f = (1, 1, 1, 0)$, $w(f) = 3$ y $d(f, g) = w(f \oplus g) = 4$.

También podemos representar una función booleana en $\mathfrak{B}_n$ por medio un polinomio en n variables. Por ejemplo, la función $f(x_1, x_2) = x_1^2 x_2 \oplus 1$ es una expresión polinomial de la función f del ejemplo anterior. Pero no es obvio que cada función booleana pueda tener una representación polinomial y que además sea única, ya que, también el polinomio $p(x_1, x_2) = x_1 x_2 \oplus 1$ representa a f. El siguiente teorema afirma que cada $f \in \mathfrak{B}_n$ tiene una representación polinomial única.

Teorema 2. *El anillo $\mathfrak{B}_n$ es isomorfo al anillo cociente $\mathbb{F}_2[x_1, \ldots, x_n]/I$, donde $I = \langle x_1^2 \oplus x_1, \ldots, x_n^2 \oplus x_n \rangle$ es el ideal en $\mathbb{F}_2[x_1, \ldots, x_n]$ generado por los polinomios $x_1^2 \oplus x_1, \ldots, x_n^2 \oplus x_n$.*

Demostración. Sea $\phi : \mathbb{F}_2[x_1, x_2, \ldots, x_n]/I \to \mathfrak{B}_n$ dada por $(p+I) \mapsto f$, donde $f : \mathbb{F}_2^n \to \mathbb{F}_2$ es tal que $\alpha \mapsto p(\alpha)$. Entonces ϕ sería una función, es decir, estaría bien definida si demostramos que para representantes diferentes, p y p', del mismo elemento de $\mathbb{F}_2[x_1, x_2, \ldots, x_n]/I$, la igualdad $\phi(p) = \phi(p')$ es verdadera.

Supongamos que $p + I = p' + I$ en $\mathbb{F}_2[x_1, x_2, \ldots, x_n]/I$, entonces $p \oplus p' \in I$ y por lo tanto $p \oplus p' = \bigoplus_{i=1}^n h_i(x_i^2 \oplus x_i)$, donde $h_i \in \mathbb{F}_2[x_1, x_2, \ldots, x_n]$ para todo $i = 1, \ldots, n$. Sea $\alpha \in \mathbb{F}_2^n$; por la definición de la función ϕ tenemos que $\phi(p \oplus p') = f$, donde $f(\alpha) = (p \oplus p')(\alpha) = p(\alpha) \oplus p'(\alpha) = 0$ para todo $\alpha \in \mathbb{F}_2^n$; de aquí que $p(\alpha) = p'(\alpha)$ para todo $\alpha \in \mathbb{F}_2^n$. Entones $\phi(p) = \phi(p')$ y por lo tanto ϕ está bien definida.

De acuerdo a la definición de ϕ, es fácil ver que ϕ es un homomorfismo de anillos. (También ϕ es una transformación lineal entre estos $\mathbb{F}_2$-espacios vectoriales.) Así que, para demostrar que ϕ es inyectiva basta mostrar que $\ker \phi = I$.

Es claro que $I \subseteq \ker \phi$. Ahora, sea $(p + I) \in \ker \phi$, entonces $\phi(p + I) = 0$; es decir, $p(\alpha) = 0$ para todo $\alpha \in \mathbb{F}_2^n$. Sea $>$ cualquier orden monomial sobre $\mathbb{Z}_{\geq 0}^n$. Dividamos al polinomio p, de acuerdo al algoritmo de la división en $K[x_1, \ldots, x_n]$, entre la n-ada ordenada de polinomios $F = (x_1^2 \oplus x_1, x_2^2 \oplus x_2, \ldots, x_n^2 \oplus x_n)$. Entonces

$$p = \bigoplus_{i=1}^{n} h_i(x_i^2 \oplus x_i) \ \oplus r(x_1, x_2, \ldots, x_n),$$

donde ningún x_i^2 divide a ningún monomio del polinomio $r(x_1, x_2, \ldots, x_n)$ ya que $\mathrm{TP}(x_i^2 \oplus x_i) = x_i^2$ (ver el lema 1). Por lo tanto, los monomios de r son de la forma x_i ó $x_{i_1} x_{i_2} \cdots x_{i_k}$ para algún $k \in \mathbb{Z}$. Ya que $\bigoplus_{i=1}^n h_i(x_i^2 \oplus x_i) \in I$ y $p(\alpha) = 0$, tenemos que $r(\alpha) = 0$ para todo $\alpha \in \mathbb{F}_2^n$. Si el monomio x_i aparece en el polinomio r, entonces evaluando r en el vector $e_i \in \mathbb{F}_2^n$, obtenemos precisamente el coeficiente del monomio x_i en r, digamos a_i. Como $r(e_i) = 0$, vemos que $a_i = 0$. Así, el polinomio r no tiene monomios de la forma x_i. De manera similar, si algún monomio de la forma $x_{i_1} x_{i_2}$ aparece en el polinomio r, entonces evaluando r en el vector $\alpha \in \mathbb{F}_2^n$ que tiene unos en las coordenadas i_1 e i_2 y ceros en las restantes, obtenemos como resultado el coeficiente del monomio $x_{i_1} x_{i_2}$, digamos a_{i_1, i_2}. Nuevamente, como $r(\alpha) = 0$, tenemos que $a_{i_1, i_2} = 0$. Procediendo de esta manera, vemos que r es el polinomio cero. Por lo tanto, $p = \bigoplus_{i=1}^n h_i(x_i^2 \oplus x_i)$; o sea, $p \in I$ y de esta forma

hemos demostrado que $\ker \phi \subseteq I$. Entonces, $I = \ker \phi$, es decir, ϕ es inyectiva.

Por otro lado, sea $f \in \mathfrak{B}_n$. Necesitamos un elemento $p + I \in K[x_1, \ldots, x_n]/I$ tal que $\phi(p + I) = f$. Para construir tal elemento usemos el siguiente algoritmo:

1. *Entrada: La función booleana* $f \in \mathfrak{B}_n$.

2. *Haga* $p_0(x_1, x_2, \ldots, x_n) = f(0, 0, \ldots, 0)$.

3. *Para* $k = 1$ *a* $2^n - 1$ *haga*
calcule la representación binaria del entero k,
$k = b_0 + b_1 2 + b_2 2^2 + \cdots + b_{n-1} 2^{n-1}$.
Si $p_{k-1}(b_0,\ b_1, \ldots, b_{n-1}) \neq f(b_0, b_1, \ldots, b_{n-1})$ *entonces*
$p_k(x_1, x_2, \ldots, x_n) = p_{k-1}(x_1, x_2, \ldots, x_n) \oplus \prod_{i=1}^{n} (x_i^{b_{i-1}})$.
De otra manera,
$p_k(x_1, x_2, \ldots, x_n) = p_{k-1}(x_1, x_2, \ldots, x_n)$.

4. *Salida:* $p_{2^n-1}(x_1, x_2, \ldots, x_n)$.

Afirmamos que $p = p_{2^n-1}(x_1, x_2, \ldots, x_n)$ es tal que $\phi(p + I) = f$, es decir, para todo $\alpha \in \mathbb{F}_2^n$ tenemos que $p(\alpha) = f(\alpha)$. Para demostrar lo anterior, supongamos que estamos en el j-ésimo paso del algoritmo, $1 \leq j \leq 2^n - 1$, y que $p_{j-1} = f$ para toda representación binaria de k con $0 \leq k \leq j - 1$. Si en este paso no se suma ningún monomio, entonces $p_j = p_{j-1}$ y por lo tanto $p_j = f$ para toda representación binaria de k con $0 \leq k \leq j$. Por el contrario, si añadimos el monomio $\prod_{i=1}^{n} x_i^{b_{i-1}}$, donde $(b_0, b_1, \ldots, b_{n-1})$ es la representación binaria de j, sean $b_{i_1}, \ldots, b_{i_s}$ las coordenadas diferentes de cero del vector $(b_0, b_1, \ldots, b_{n-1})$. El monomio $x_{i_1}^{b_{i_1}} \cdots x_{i_s}^{b_{i_s}}$ al ser evaluado en las representaciones binarias de los enteros menores estrictos que j es cero; si esto no fuese así, entonces existe un vector $v \in \mathbb{F}_2^n$ correspondiente a la representación binaria de un entero no negativo menor estricto que j, el cual tiene $1'$s en las posiciones $i_1, \ldots, i_s$ y probablemente en otras. Pero esto implica que este entero es mayor o igual que j, lo cual no es posible. Entonces $p_j = p_{j-1} = f$ para toda representación binaria de k con $0 \leq k < j$. Ahora, el monomio $x_{i_1}^{b_{i_1}} \cdots x_{i_s}^{b_{i_s}}$ es 1 cuando es evaluado en la representación binaria de j, lo cual hace que p_j coincida con f en la representación binaria de j. Por lo tanto, $p_j = f$ para toda representación binaria de k con $0 \leq k \leq j$. □

Este teorema nos dice que toda función booleana $f \in \mathfrak{B}_n$ tiene una

única representación polinomial, módulo el ideal I,

$$f(x_1, x_2, \ldots, x_n) = \bigoplus_{u \in \mathbb{F}_2^n} \lambda_u \left(\prod_{i=1}^{n} x^{u_i} \right), \quad \lambda_u \in \mathbb{F}_2, \quad u = (u_1, u_2, \ldots, u_n).$$

Esta representación de f es llamada la *forma normal algebraica* (F.N.A.) de f. El grado total $\mathrm{gr}(f)$ de la F.N.A. es llamado el *grado algebraico* de la función.

Una desventaja de la tabla de verdad es que no da información acerca del grado algebraico de la función y sobre el número de términos de f en su F.N.A.. La tabla de verdad puede ser recuperada de la F.N.A. y recíprocamente, la F.N.A. puede ser calculada a partir de la tabla de verdad por medio del algoritmo dado en la prueba del teorema 2, como se muestra en el ejemplo siguiente.

Ejemplo 4. Para cada $k \in \{0, 1, , \ldots, 7\}$ calcule su representación binaria $k = b_0 + 2b_1 + 4b_2$ y sea $v_k = (b_0, b_1, b_2)$ el vector asociado a k. Considere la función $f \in \mathfrak{B}_3$ dada por la tabla de verdad $V_f = (0, 1, 1, 1, 0, 1, 0, 1)$. Para calcular la F.N.A. de f primero hacemos $p_0(x_1, x_2, x_3) = f(0, 0, 0) = 0$. Luego, comparamos los valores de p_0 y f en v_1. Vemos que $p_0(v_1) = 0$ y $f(v_1) = 1$, no coinciden. De acuerdo al algoritmo hacemos

$$p_1(x_1, x_2, x_3) = p_0(x_1, x_2, x_3) \oplus x_1^1 x_2^0 x_3^0 = 0 \oplus x_1 = x_1.$$

Ahora comparamos p_1 y f en v_2. Ya que $p_1(v_2) = 0$ y $f(v_2) = 1$ son diferentes tenemos que añadir un término a p_1 para obtener p_2:

$$p_2(x_1, x_2, x_3) = p_1(x_1, x_2, x_3) \oplus x_1^0 x_2^1 x_3^0 = x_1 \oplus x_2 = x_1 \oplus x_2.$$

Resumimos el procedimiento en la siguiente tabla.

k	$k = b_0 + b_1 2 + b_2 2^2$	$f(b_0, b_1, b_2)$	$p_k(x_1, x_2, x_3)$
0	(0, 0, 0)	0	0
1	(1, 0, 0)	1	$0 \oplus x_1$
2	(0, 1, 0)	1	$x_1 \oplus x_2$
3	(1, 1, 0)	1	$x_1 \oplus x_2 \oplus x_1 x_2$
4	(0, 0, 1)	0	$x_1 \oplus x_2 \oplus x_1 x_2$
5	(1, 0, 1)	1	$x_1 \oplus x_2 \oplus x_1 x_2$
6	(0, 1, 1)	0	$x_1 \oplus x_2 \oplus x_1 x_2 \oplus x_2 x_3$
7	(1, 1, 1)	1	$x_1 \oplus x_2 \oplus x_1 x_2 \oplus x_2 x_3 \oplus x_1 x_2 x_3$

Por lo tanto, la F.N.A. de f es $p(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_1 x_2 \oplus x_2 x_3 \oplus x_1 x_2 x_3$. El $\mathrm{gr}(f) = 3$ y el número de términos en su forma normal algebraica es 5.

Referencias

[1] C. Carlet; P. Guillot. *A New Representation of Boolean Functions.* Lecture Notes In Computer Science; Vol. 1719. Proceedings of the 13th International Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes; pp. 94 - 103. 1999.

[2] D. Cox; J. Little; D. O'shea. *Ideals, Varieties, and Algorithms. An Introduction to Computacional Algebraic Geometry and Commutative Algebra.* Tercera Edición. Springer. New York. 2007

[3] F. J. Macwilliams; N. J. A. Sloane. *The Theory of Error-Correcting Codes.* North-Holland. 1997