Discrete Applied Mathematics 186 (2015) 1–6

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Recursion orders for weights of Boolean cubic rotation symmetric functions

Thomas W. Cusick *, Bryan Johns

University at Buffalo, 244 Math. Bldg., Buffalo, NY 14260, United States

ARTICLE INFO

Article history:
Received 1 July 2014
Received in revised form 7 January 2015
Accepted 9 January 2015
Available online 2 February 2015

Keywords:
Boolean functions
Rotation symmetry
Cubic function
Hamming weight
Recursion
Affine equivalence

ABSTRACT

Rotation symmetric (RS) Boolean functions have been extensively studied in recent years because of their applications in cryptography. In cryptographic applications, it is usually important to know the weight of the functions, so much research has been done on the problem of determining such weights. Recently it was proved that for cubic RS functions in n variables generated by a single monomial, the weights of the functions as n increases satisfy a linear recursion. Furthermore, explicit methods were found for generating these recursions and the initial values needed to use the recursions. It is important to be able to compute the order of these recursions without needing to determine all of the coefficients. This paper gives a technique for doing that in many cases, based on a new notion of towers of RS Boolean functions.

1. Introduction

Boolean functions have a variety of applications in the field of cryptography, a thorough overview of which can be found in [10]. A Boolean function in n variables can be defined as a map from $\mathbb{V}_n$, the n-dimensional vector space over the two element field $\mathbb{F}_2$, to $\mathbb{F}_2$. If f is a Boolean function in n variables, the *truth table* of f is defined to be the 2^n-tuple given by $(f(\mathbf{v}_0), f(\mathbf{v}_1), \ldots, f(\mathbf{v}_{2^n-1}))$ where $\mathbf{v}_0 = (0, \ldots, 0, 0)$, $\mathbf{v}_1 = (0, \ldots, 0, 1), \ldots, \mathbf{v}_{2^n-1} = (1, \ldots, 1, 1)$ are the 2^n elements of $\mathbb{V}_n$ listed in lexicographical order. The *weight* or *Hamming weight* of f (notation $wt(f)$) is the number of 1's that appear in the truth table of f. In cryptographic applications of Boolean functions it is usually important to know the weight of the functions, so much research (see the book [10] for many references) has been done on the problem of determining such weights.

As described in [10, p. 6], every Boolean function on $\mathbb{V}_n$ can be expressed as a polynomial over $\mathbb{F}_2$ in n binary variables by:

$$f(x_1, \ldots, x_n) = \sum_{\mathbf{a} \in \mathbb{V}_n} c_{\mathbf{a}} x_1^{a_1} \cdots x_n^{a_n}$$

where $c_{\mathbf{a}} \in \mathbb{F}_2$ and $\mathbf{a} = (a_1, \ldots, a_n)$ with each a_i equal to 0 or 1. The above representation is referred to as the *algebraic normal form* (ANF) of f. Let d_i be the number of variables in the ith monomial of f, so d_i is the *algebraic degree* (or just the *degree*) of the monomial. If we let D be the set of the distinct degrees of the monomials in f which have non-zero coefficients, then the *degree* of f is given by $\max(D)$. If D contains only one element, then each monomial in f has the same degree and f is said to be *homogeneous*. If the degree of f is 1, then f is said to be *affine*, and if f is affine and homogeneous (i.e. the constant term is 0), f is said to be *linear*.

A Boolean function f is said to be *rotation symmetric* if its ANF is invariant under any power of the cyclic permutation $\rho(x_1, \ldots, x_n) = (x_2, \ldots, x_n, x_1)$. We use the notation $(1, r, s)_n$, as in [5], for the cubic function in n variables generated

* Corresponding author.
E-mail address: cusick@buffalo.edu (T.W. Cusick).

http://dx.doi.org/10.1016/j.dam.2015.01.017
0166-218X/

by the monomial $x_1x_rx_s$, and we shall call such a function a cubic *MRS function* (short for monomial rotation symmetric function). Note that we do not always assume $r < s$ and that (except for the *short function* $(1, \frac{n}{3}+1, \frac{2n}{3}+1)$ where 3 divides n, which has only $n/3$ monomials in its ANF) we always have

$$(1, r, s)_n = (1, n-r+1, n+s-r+1)_n = (1, n-s+1, n+r-s+1)_n$$

because every variable appears three times in the n monomials in the ANF. If we are free to choose a representation $(1, r, s)_n$ for a function, we will always take $r < s$ and choose the least value of r; however, in various proofs below we will sometimes need to use other representations for a function.

Rotation symmetric functions have been extensively studied in the last fifteen years because they have many applications in cryptography. Some recent papers in this area of research are [4,12–15,17–19].

We shall use the notation $[i, j, k]$ for the monomial $x_ix_jx_k$. Unless otherwise specified, all subscripts in given monomials will be taken Mod (n) (where the capital Mod notation i Mod (n) indicates that i is reduced modulo n and $i \in \{1, 2, \ldots, n\}$).

Let $\sigma(f)$ denote a permutation of the variables in the function f. If, given any rotation symmetric function f, $\sigma(f)$ is also rotation symmetric, we say σ *preserves rotation symmetry*. Also, without loss of generality, we assume that if σ maps $(1, r, s)$ to $(1, p, q)$, then $\sigma([1, r, s]) = [1, p, q]$ (if $\sigma([1, r, s]) = [i, j, k]$, where $[i, j, k]$ is another monomial term in 2-$(1, p, q)$, then we could take a map β that decreases the index of each variable by $i - 1$ Mod$(2n)$ and consider instead $\sigma' = \beta \circ \sigma$).

Two Boolean functions f and g in n variables are said to be *affine equivalent* if there exists an invertible matrix A with entries in $\mathbb{F}_2$ and $\mathbf{b} \in \mathbb{V}_n$ such that $f(\mathbf{x}) = g(A\mathbf{x} \oplus \mathbf{b})$. Throughout the paper we shall use $f \sim g$ to mean that the Boolean functions f and g are affine equivalent. In general, determining whether or not two Boolean functions are affine equivalent is difficult, even in the simplest cases. Recently, however, much work has been done on affine equivalence of MRS functions (see [2,5,8,7,16]). In particular, [16] determines all of the affine equivalence classes for quadratic MRS functions, [5] determines all of the affine equivalence classes under permutations for the cubic MRS functions and [8] determines all of the affine equivalence classes under permutations for the quartic MRS functions.

In 2012, Bileschi, Cusick and Padgett [1] gave an explicit algorithm for finding a linear recursion for the sequence $\{wt((1, r, s)_n) : n = s, s+1, \ldots\}$ where $(1, r, s)_n$ is any cubic MRS function. The method of [1] was greatly simplified by Brown and Cusick [3]. We let $d(r, s)$ denote the degree of the characteristic polynomial for the recursion (see for example [11, p. 1]; of course this degree is usually called the *order* of the recursion). For brevity, we shall refer to the characteristic polynomial for the recursion for the sequence of weights $wt((1, r, s)_n)$ as the *recursion polynomial* for $(1, r, s)$.

These recursions can be efficiently computed, as is explained in Section 2, as long as s is not too large. Even the computation of the initial values of the function weights, which is needed to compute weights for values of n beyond the initial values, can be carried out efficiently, without the need to look at large truth tables. This is explained in Remark 1 in Section 4.

The problem of getting information about the recursion orders $d(r, s)$ as a function of r and s was raised in [1, Section 9]. In this paper we give a method for obtaining such information based on a result of Cusick and Padgett [9, Corollary 3.2]; this is explained in Section 2, which also introduces the convenient new concept of *towers* of cubic MRS functions.

2. Towers of rotation symmetric functions

For the rest of the paper, we shall use two notations for an n-dimensional vector of variables, namely

$$\mathbf{x} = (x_1, x_2, \ldots, x_n)$$

and

$$\mathbf{y} = (x_0, x_1, \ldots, x_{n-1}).$$

The $\mathbf{y}$ notation is much more convenient when using some of the results from [9]. For example, we use the notation $(0, r-1, s-1)_n$, as in [9], for the cubic MRS function in n variables generated by the monomial $x_0x_{r-1}x_{s-1}$. Of course this is the same function $f(\mathbf{x})$ that we previously labeled with the notation $x_1x_rx_s$.

Given a rotation symmetric function $f(x_1, \ldots, x_n) = (1, r, s)_n$ in n variables generated by $x_1x_rx_s$ such that $r < s$ and $\gcd(r-1, s-1) = 1$, we define the *(r, s)-tower* of functions $F_1, F_2, \ldots$ by

$$F_k = (1, k(r-1)+1, k(s-1)+1)_n, \quad k = 1, 2, \ldots.$$

We will simply refer to this sequence of functions as a *tower* when the defining pair (r, s) is understood. If the value of n for the function needs to be specified, we use the longer notation $F_{k,n}$ instead of F_k. We also think of the functions $F_{k,n}$ for $k = 2, 3, \ldots$ as being arranged in order above the base of the tower, so functions with larger k are higher up in the tower. When we consider some function F_k in the tower, we assume n is large enough $(n \geq k(s-1)+1)$ so that the function is defined. Since $k \leq (n-1)/(s-1)$, every tower has finitely many functions, but the height (that is, number of functions) of the towers increases as n increases. We will sometimes refer to the k in $F_{k,n}$ as the *k-value* of F. We shall make extensive use of the following result [9, Corollary 3.2] which says that for any given integer k, the weight of any function F_k in the tower with $n = km$ variables can be computed from the weight of the function $F_1 = (1, r, s)_n$, which we say is the *base* of the tower with n variables.

Lemma 1. *Given integers r and s with $1 < r < s$ and an integer $k > 1$, we have the following relation between the weights of the given cubic MRS functions:*

$$wt((1, k(r-1)+1, k(s-1)+1)_{kn}) = \frac{1}{2}\left(2^{kn} - (2^n - 2wt((1, r, s)_n))^k\right). \tag{1}$$