AIMS Mathematics, 9(5): 10609–10637.
DOI: 10.3934/math.2024518
Received: 18 February 2024
Revised: 11 March 2024
Accepted: 12 March 2024
Published: 18 March 2024

http://www.aimspress.com/journal/Math

Research article

Identifying codewords in general Reed-Muller codes and determining their weights

Claude Carlet[1,2,*]

[1] Department of Mathematics, University of Paris 8, 93526 Saint-Denis, France
[2] Department of Informatics, University of Bergen, 5005 Bergen, Norway

* **Correspondence:** Email: claude.carlet@gmail.com.

Abstract: Determining the weight distribution of all Reed-Muller codes is a huge and exciting problem that has been around since the sixties. Some progress has been made very recently, but we are still far from a solution. In this paper, we addressed the subproblem of determining as many codeword weights as possible in Reed-Muller codes of any lengths and any orders, which is decisive for determining their weight spectra (i.e., the lists of all possible weights in these codes). New approaches seem necessary for both the main problem and the subproblem. We first studied the difficulties and the limits of the approach, which consisted of using the usual primary and secondary constructions of Boolean functions for the purpose of determining as many weights as possible in Reed-Muller codes. We then introduced a way, different from the usual constructions, to generate Boolean functions in n variables having an algebraic degree bounded from above, without any restriction on n, and whose Hamming weights can be determined. This provided weights in Reed-Muller codes of any lengths 2^n and any orders, allowing us to reach potentially new values in the weight spectra of Reed-Muller codes (as we illustrate with all Reed-Muller codes of lengths up to 2^{21}), with the related codewords being given with their supports and their algebraic normal forms being mathematically derived.

Keywords: Reed-Muller codes; weight spectrum
Mathematics Subject Classification: 94B05, 94C10

1. Introduction

For every nonnegative integers r, n such that $r \leq n$, the Reed-Muller code $RM(r, n)$ of length* $N = 2^n$ and order r equals the vector space over $\mathbb{F}_2$ of n-variable Boolean functions of algebraic degree at most r. Recall that each n-variable Boolean function $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ admits a representation in the form of a

*Usually in coding theory, the length of a code is denoted by n, but since we deal with Boolean functions, we keep n for the number of variables; we denote then the length by N.

multivariate polynomial over $\mathbb{F}_2$ of a particular shape:

$$f(x) = \sum_{I \subseteq \{1,\ldots,n\}} a_I x^I; \quad a_I \in \mathbb{F}_2,\ x = (x_1, \ldots, x_n) \in \mathbb{F}_2^n,\ x^I = \prod_{i \in I} x_i \tag{1.1}$$

(the sum being calculated modulo 2). Such representation is unique for each function and is called its algebraic normal form (ANF). The global degree $\max\{|I|; a_I = 1\}$ of the ANF is called the algebraic degree of f.

For a binary block code needing to be a subset of $\mathbb{F}_2^N$ for some N, each Boolean function is identified with the list of its $N = 2^n$ values, some order on $\mathbb{F}_2^n$ being previously chosen. When we shall speak of codewords of Reed-Muller codes, we will not make the difference between an n-variable Boolean function and the corresponding vector of length N.

Reed-Muller codes were introduced in 1954 by David Muller in [27] and their decoding algorithm was given the same year by Irving Reed in [29]. These codes have originally played an important role in the theory of error correcting codes, as well as in their applications. It is well known that the Reed-Muller code $RM(1, 5)$ was used in the sixties for correcting the errors of transmission of the first photographs of Mars by Mariner. These photographs were in black and white. Every codeword corresponded to the level of brightness of a pixel. There were 64 different levels since there are 64 codewords in $RM(1, 5)$, and the minimum distance of this code was equal to 16, with up to $\left\lfloor \frac{16-1}{2} \right\rfloor = 7$ errors that could be corrected in the transmission of each codeword[†].

Reed-Muller codes were also used in the 3rd generation (3G) of mobile phones (starting in 2000). Reed-Muller codes intervened in the initial "handshake" between the mobile device and the base station, whose role was to inform the receiver of what type of communication would come next. Again, $RM(1, 5)$ was initially used for this purpose, and it was later replaced by a punctured subcode of the second-order Reed-Muller code $RM(2, 5)$, which had a dimension of 10 and a minimum distance of 12.

The parameters of Reed-Muller codes are not so good, except for the first order, but they contain optimal codes such as the Kerdock codes [19]. They still play an important role nowadays, thanks to their specific properties (see, e.g., [2, 13]) and their roles with respect to new problematics, such as locally correctable codes [20]), low degree testing, private information retrieval, and compressed sensing. The interest in Reed-Muller codes has also been renewed because of polarization (see, e.g., [24]). At various block-lengths and rates, Reed-Muller codes can be superior to polar codes [25], even for 5G [14]. A nice survey on Reed-Muller codes can be found in [1].

We can easily generate the ANF (1.1) of (infinite classes of) codewords in any Reed-Muller codes, but in most cases, it is impossible to calculate (mathematically) their Hamming weight $w_H(f) = |\{x \in \mathbb{F}_2^n; f(x) = 1\}|$.

Determining Hamming weights (if possible, all weights of codewords, and, if possible, the whole weight distribution) in Reed-Muller codes has always been considered very important; see, e.g., the papers [4, 5, 7, 12, 15–18, 23, 26, 30], the data in [31], and the books [22, 28]. The weight distributions of the Reed-Muller codes of length 2^n and orders $0, 1, 2, n-2, n-1, n$ are known. The weights in these codes equal $0, 2^n$ for the order 0, with additionally 2^{n-1} for the order 1, and $2^{n-1} \pm 2^i$ where $\frac{n}{2} \leq i \leq n$ for the order 2; see, e.g., [22]. The weights in $RM(n, n)$ are all integers between 0 and 2^n since $RM(n, n) = \mathbb{F}_2^{2^n}$; the weights in $RM(n-1, n)$ are all even integers between 0 and 2^n; the weights in

[†]In the late seventies, for transmitting color photographs of Mars, Voyager used the extended binary Golay code and still later Reed-Solomon codes

$RM(n-2,n)$ are all even integers between 0 and 2^n except 2 and 2^n-2. For all these codes, the weight distributions are known (thanks to the Mac Williams identity for the orders $n-2, n-1$ [21, 22], since the dual of $RM(r,n)$ equals $RM(n-r-1,n)$). The weight distributions of some Reed-Muller codes $RM(r,m)$ have been determined thanks to heavy computations, for m small enough; they are reported in [31].

The weights in $RM(n-3,n)$ have been recently determined in [12]. They are all even integers in $\{0,2,4,...,2^n\} \setminus \{2,4,6,10,2^n-10,2^n-6,2^n-4,2^n-2\} = \{0,8,12+2i,2^n-12,2^n-8,2^n\}$, where i ranges over consecutive integers from 0 to $2^{n-1}-13$. They have been obtained by an induction (the Mac Williams identity does not allow us to determine the weight distribution, which is still unknown despite the fact that the weight distribution of $RM(2,m)$ is known, because the expression of the number of codewords of Hamming weight 2^{n-1} in $RM(2,n)$ is too complex). This induction does not allow us to determine the weight distribution, and new ideas to be found seem necessary for obtaining it. However, determining the weight spectrum[‡] of $RM(n-3,n)$ is already a step forward.

The same method worked for determining the weights in $RM(n-4,n)$ [12], which are all integers in $\{0,16,24,28+2i,2^n-28,2^n-24,2^n-16,2^n\}$, where i ranges over the set of consecutive integers from 0 to $2^{n-1}-29$. The weights in $RM(n-5,n)$ could not be determined in [12], but they were found in [9]; they are all integers in $\{0,32,48,56,60,62,64,68,72+2i,2^n-68,2^n-64,2^n-62,2^n-60,2^n-56,2^n-48,2^n-32,2^n\}$, where i ranges over the set of consecutive integers from 0 to $2^{n-1}-72$.

For general Reed-Muller codes, bounds are known on the weight enumerators, which are useful for studying the capacity of Reed-Muller codes on the binary erasure channel and the binary symmetric channel (see [1, Chapter 4]), but our knowledge on the weights themselves is limited.

McEliece's theorem [23] shows that the weights in $RM(r,n)$ are divisible by $2^{\lfloor \frac{n-1}{r} \rfloor}$, and Kasami-Tokura's result (that we shall recall in Section 2) and Kasami-Tokura-Azumi's results [17] give the weights of $RM(r,n)$, which are between the minimum distance $d = 2^{m-r}$ and 2.5 times d. It is conjectured in [12] that for every constant c and for n large enough, the weight spectrum of $RM(n-c,n)$ is made of 0 and 2^n and all the weights between the minimum distance 2^c and its complement to the length 2^n, which are authorized by McEliece's theorem and Kasami-Tokura-Azumi's results. This means, in particular, that every even number between 2.5 times the minimum distance and its complement to 2^m would be a weight in $RM(m-c,m)$. This conjecture[§] is verified by the weight spectra of $RM(n-5,n), RM(n-4,n)$ and $RM(n-3,n)$. The method used in [9, 12] for handling these three weight spectra is the same: There is a corollary in [30], which can easily be proved directly, and which says that the weight spectrum of $RM(r,n)$ includes $A+A$, where A is the weight spectrum of $RM(r-1,n-1)$. This allows us to address the weight spectrum of $RM(n-c,n)$ by an induction on n, starting from a value n_0 such that the weight spectrum of $RM(n_0-c,n_0)$ is already generic, which means that it has, according to McEliece's theorem, a divisibility by 2 and not by a larger power of 2. This means that we need to start from $n_0 \geq 2c$. Indeed, according to McEliece's theorem, all the weights in $RM(c-1,2c-1)$ are divisible by 4, while those in $RM(c,2c)$ are divisible by 2. We know from [6] that McEliece's divisibility bound is tight in the sense that there is at least a codeword in every $RM(r,n)$ code, with a weight congruent to $2^{\lfloor \frac{n-1}{r} \rfloor}$ modulo $2^{\lfloor \frac{n-1}{r} \rfloor+1}$. We can try to see whether the weights obtained from $A+A$, where A is the weight spectrum of $RM(c,2c)$, allow us to reach all the weights authorized by

[‡]In coding theory, contrary to Boolean function theory, the spectrum does not include the multiplicities of the values (when these multiplicities are taken into account, we speak of weight distribution).

[§]It seems a little risky to present this as a conjecture and in [9], it is then presented as an open question.

McEliece's theorem and Kasami-Tokura-Azumi's result. The first difficulty is then to reach all weights in $RM(c, 2c)$. In the case of $c = 3, 4$, this has been rather easy, but proving the conjecture recalled above for $c = 5$ with this method, which needs to start the induction with $n = 10$ (a value much larger than what can be reached with the heavy computations made by M. Terada, J. Asatani, and T. Koumoto and reported in [31]), has led to the construction of functions in 10 variables with an algebraic degree of at most 5 and having all possible even weights between 2.5 times the minimum distance 32, that is, 80 and $2^{10} - 80$. The next step $c = 6$ needs to address the code $RM(6, 12)$, which has huge parameters [4096,2510], while the largest reached currently are [512,256] and [512,382]). It is shown in [9] how determining the weight spectrum of $RM(6, 12)$ needs to determine whether some specific values (such as 166), which are "holes" after general methods were applied, are the weights of codewords. This may not be as hard as expected for $c = 6$, but addressing larger values of c will probably lead to more of such "holes". Hence, being able to build as many weights as possible in Reed-Muller codes is of a great importance, and in particular, reaching weights that are not obtained by classic constructions.

Providing weights can indeed be tried by investigating the known (primary and secondary) constructions of Boolean functions and deducing functions whose weight can be determined, as was done in [9]. Some weights are easily reached this way, but we can expect that these constructions will not suffice for addressing the weights in $RM(n - c, n)$ for larger values of c.

Note that the codes $RM(n-c, n)$ considered above, being such that $n \geq 2c$, are of the form $RM(r, n)$ with $n \geq 2(n - r)$, that is, $r \geq \frac{n}{2}$. Another case where more weights in Reed-Muller codes $RM(r, n)$ are useful information is when $r < \frac{n}{2}$.

Recall that when Boolean functions in n variables are given, for instance, by their ANF, with n ranging over $\mathbb{N}$, it is rarely possible to mathematically evaluate their Hamming weights. Of course, it is always possible when the function is affine (belonging then to the Reed-Muller code of order 1), but this provides only three weights for each n. When the function is taken quadratic (i.e., belonging to the second-order Reed-Muller code), there are methods for determining its weight (see a survey in [8, Chapter 4]). However, these methods allow us to concretely address only a few cases (even the first step, which consists of determining the linear kernel of the function, is impossible to complete systematically). The weights of quadratic functions are very specific. The indicators of affine spaces (flats) are also addressable, but their weights are minimal in the Reed-Muller codes to which they belong. It needs specific work to study the weights of Boolean functions obtained by the constructions evoked above, and we shall describe in Section 2, as nothing automatic exists.

The problem we want to address in this paper is not as hard as determining the weight of any given Boolean function: We only want to find as many weights as possible in general Reed-Muller codes. However, it is not so easy to provide codewords of Reed-Muller codes whose weights can be determined.

For finding more weights, methods complementary to the usual constructions are needed. In the present paper, we give such a method to automatically generate codewords in Reed-Muller codes of any lengths 2^n. These codewords depend on the number of variables n, the order r, a parameter t, and the choice of t vectors a_i. We have, thanks to a property of the corresponding functions, an upper bound on their algebraic degree (but determining the degree exactly would be difficult, and even trying to directly show this upper bound by working on the ANF of the functions seems quite hard). The weights of these functions can be evaluated or at least bounded from above, because when these Boolean functions are given as the sums (modulo 2) of atomic ones, the only limitation for evaluating

their weights is to determine the number of these atomic functions which appear an odd number of times in the expression.

There is a case (when the vectors a_i involved in the construction are linearly independent) where we can ensure that all these atomic functions are distinct, which allows us to exactly calculate the Hamming weight. This provides information on the weight spectra of Reed-Muller codes when they are unknown (that is, currently, for the orders from 3 to $n-6$). For instance, we shall see in the tables provided that our method gives weights in $RM(r,n)$ that are much larger than twice the minimum distance and have low valuation.

The case mentioned above, where the vectors a_i are linearly independent, provides at most $\frac{n}{2}$ distinct weights for each Reed-Muller code, and this is not much. We then investigate two cases where the vectors are linearly dependent. We do not cover all the cases where the vectors are linearly dependent (it seems impossible to do so), but other cases could be similarly investigated.

We also study the weights of the sums of the designed functions, in a case where we know they have disjoint supports. This provides many more weights.

The paper is structured as follows. In Section 2, we recall the state of the art in the determination of weights in Reed-Muller codes by using the classic constructions (Maiorana-McFarland, etc.). We show the difficulties presented by this method and why it suits better for low orders. In Section 3, we introduce our new construction of Reed-Muller codewords and we study some particular cases. We determine the weights under a condition that is rather general (namely, some vectors a_i involved in the construction are linearly independent), and we also study two cases where this condition is not satisfied; this provides a list of weights for each Reed-Muller code, which is longer for larger orders. We then show that more weights - a huge number when the order is large enough - can be obtained as the additions of some of these weights. To conclude this section, we determine the ANF of the constructed functions when the vectors a_i are linearly independent. We conclude with some observations on future work.

2. State of the art on the Hamming weights of Reed-Muller codewords

It is well-known that the minimum nonzero Hamming weight of $RM(r,n)$ equals 2^{n-r} (see [22, Chapter 13], and see [8, Chapter 4] for a more direct proof), and that the nonzero minimum weight codewords in this code are the indicators of the $(n-r)$-dimensional affine subspaces of $\mathbb{F}_2^n$.

All the low Hamming weights are known in all Reed-Muller codes, and there are very few: Berlekamp and Sloane [4] (see the Addendum in this paper) and Kasami and Tokura [16] have shown that, for $r \geq 2$, the only Hamming weights in $RM(r,n)$ occurring in the range $[2^{n-r}; 2^{n-r+1}[$ are of the form $2^{n-r+1} - 2^{n-r+1-i}$, where we have $i \leq \max(\min(n-r,r), \frac{n-r+2}{2})$. The latter has completely characterized the codewords: The corresponding functions are affinely equivalent either to $x_1 \cdots x_{r-2}(x_{r-1}x_r + x_{r+1}x_{r+2} + \cdots + x_{r+2l-3}x_{r+2l-2})$, $2 \leq 2l \leq n-r+2$, or to $x_1 \cdots x_{r-l}(x_{r-l+1} \cdots x_r + x_{r+1} \cdots x_{r+l})$, $3 \leq l \leq \min(r, n-r)$. The functions whose Hamming weights are strictly less than 2.5 times the minimum distance 2^{n-r} have later been studied in [17].

Recall that, on the contrary, the general weights in $RM(r,n)$ can be rather diverse, as soon as $r \geq 3$ and n is large enough. Indeed, as shown in [7], for every Boolean function f on $\mathbb{F}_2^n$, there exist an integer m and a Boolean function g of an algebraic degree of at most 3 on $\mathbb{F}_2^{n+2m}$, such that $\sum_{x\in\mathbb{F}_2^{n+2m}}(-1)^{g(x)} = 2^m \sum_{x\in\mathbb{F}_2^n}(-1)^{f(x)}$, which gives the following relation between the Hamming weights of f and g: $2^{n+2m} -$

$2w_H(g) = 2^m(2^n - 2w_H(f))$. Hence, the Hamming weight of f is related in a simple way to the Hamming weight of a cubic function (in a number of variables which can be exponentially larger). This shows that the weights in $RM(3, n)$ (that is, the distances) can be complex, contrary to those in $RM(2, n)$. Unfortunately, this result does not provide an efficient method for finding weights in third-order Reed-Muller codes: Trying to find new weights in these codes by starting with Boolean functions f of any degree in less variables and applying the result does not work well, because m in this result is exponentially large compared to n.

The possible weights of the codewords in the Reed-Muller codes of orders $3, \ldots, n-6$ whose values lie between $2.5\,d$ and $2^n - 2.5\,d$ are unknown[¶], except for some functions that we shall describe, and which hardly allow to provide non-peculiar weights for general Reed-Muller codes:

- Quadratic functions, in the form $f(x) = l_1(x)l_2(x) + l_3(x)l_4(x) + \cdots + l_{2k-1}(x)l_{2k}(x) + l_{2k+1}(x)$, possibly added with constant 1 (that is, complemented), when we are able to ensure that the linear functions $l_1, \ldots, l_{2k}$ are linearly independent. Then f equals the function $x_1x_2 + \cdots + x_{2k-1}x_{2k}$ composed on the right by a linear or an affine automorphism (we say that such a function is linearly, respectively affine, equivalent to $x_1x_2 + \cdots + x_{2k-1}x_{2k}$), added with an affine function (we say then that the function is extended affine equivalent to $x_1x_2 + \cdots + x_{2k-1}x_{2k}$; see more on equivalences in [8, Chapter 2]), and we can evaluate its Hamming weight. This provides weights $0, 2^{n-1}, 2^{n-1} \pm 2^i$, where $i = \lceil \frac{n}{2} \rceil$, 2^n, which are all weights in $RM(2, n)$ (all being easy to produce), but are rather peculiar in the larger Reed-Muller codes. We can also calculate the weights of the concatenations of such functions, of course, whose weights are a little more general (but the algebraic degree needs to then be determined).
- Indicators of flats (and their concatenations as well), that is, minimum nonzero weight codewords in Reed-Muller codes (see [22, Chapter 13]), in the form $\prod_{i \in I}(a_i \cdot x + \epsilon_i)$, where $a_i \in \mathbb{F}_2^n$, $\epsilon_i \in \mathbb{F}_2$, when we are able to ensure that the vectors a_i are linearly independent. This provides weights 2^i, where $i = 0, \ldots, n$, which are also easy to produce but are peculiar, too. Note that this class of functions is (as the previous one) preserved by affine equivalence.
- Functions whose weight is smaller than twice-and-a-half the minimum distance d of the Reed-Muller codes to which they belong. We have recalled above what these weights are when they are smaller than $2d$; between $2d$ and $2.5d$, the weights (determined in [17]) are too numerous for being recalled here; They are easy to produce but we encounter the same difficulty as for quadratic functions if we want to exhibit all functions with such weights: We know that they are affine equivalent to some particular functions, but ensuring such affine equivalence is not mathematically possible in an exhaustive way. Anyway, this strong result by Kasami, Tokura, and Azumi allows us to reach in Reed-Muller codes all weights smaller than 2.5 times the minimum distance (and their complements to 2^n). The question is then to find as many other weights as possible.
- Some functions obtained by using the classic primary constructions of Boolean functions, in particular, Maiorana-McFarland, Niho, and $\mathcal{PS}_{ap}$-like constructions; see [8, Chapter 4]. This allows us to reach some weights, but numerous subclasses of functions have to be separately investigated for allowing us to cover enough weights. Finding the weights that are reachable often poses technical issues, to be overcome for each subclass, such as solving equations, which can be done in some cases but not in general. To give an example, the weights of those particular

[¶]But when $n = 2r + 1$, they are known in some cases by using invariant theory, because the code is then self-dual, see [22, 28]).

Maiorana-McFarland functions of the form $f(x, y) = x \cdot \phi(y)$, where $x \in \mathbb{F}_2^t$, $y \in \mathbb{F}_2^{n-t}$, ϕ is a function from $\mathbb{F}_2^{n-t}$ to $\mathbb{F}_2^t$, and "$\cdot$" is an inner product, are deduced from the relation $\sum_{(x,y)\in\mathbb{F}_2^n}(-1)^{f(x,y)} = 2^n - 2w_H(f) = 2^t\,|\phi^{-1}(0)|$, which theoretically makes the study of the weights of these particular functions simpler. However, this replaces the difficulty of determining the weights of the functions having algebraic degrees of at most r by that of determining the possible values of the size $|\phi^{-1}(0)|$ when ϕ has an algebraic degree of at most $r-1$, that is, when all its coordinate functions have algebraic degrees of at most $r-1$. This latter problem, which is interesting to study for its own sake, may be hard since it results in determining the possible numbers of solutions of nonlinear systems of equations. Denoting the coordinate functions of ϕ by $\phi_1, \ldots, \phi_t$, the solutions of the equation $\phi(y) = 0$ are the elements of the support of the Boolean function $\prod_{i=1}^{t}(\phi_i(y) + 1)$, which has an algebraic degree of at most $t(r-1)$. In the case $t = 1$, we only get that the weights in $RM(r-1, n-1)$ are also weights in $RM(r, n)$ (which is clear since, denoting $x_i = y_{i-1}$ for $i = 2, \ldots, n$, the n-variable function $x_1 g(x_2, \ldots, x_n)$ has the same Hamming weight as the $(n-1)$-variable function g), and as soon as $t \geq 2$, the situation becomes complex. For instance, for $r = 3$ and $t = 2$, we will arrive in general to the determination of the support of a function of degree 4, which instead of reducing the degree, increases it. Moreover, the weights that are easier to obtain correspond to a large value of t and are then not quite general, since they have a valuation of at least t. The same kind of situation happens with the general Maiorana-McFarland, Niho, and $\mathcal{PS}_{ap}$-like constructions. Hence, even if it is possible to try using these classic constructions to reach weights in Reed-Muller codes, it is necessary, for reaching many weights, to have other approaches posing less problems; this is the purpose of the present paper.

- Direct sums of monomials and threshold functions (see a complete study of the cryptographic parameters of these functions in [10]). These are two cases where we can give the Hamming weights. The character sum $\sum_{x\in\mathbb{F}_2^t, y\in\mathbb{F}_2^{n-t}}(-1)^{f(x,y)}$ of a direct sum $f(x, y) = f_1(x) + f_2(y)$, of functions f_1, f_2 being the product of the character sums $\sum_{x\in\mathbb{F}_2^t}(-1)^{f_1(x)}$ and $\sum_{y\in\mathbb{F}_2^{n-t}}(-1)^{f_2(y)}$ of these functions, the Hamming weight of the direct sum $\prod_{i\in I_1} x_i + \cdots + \prod_{i\in I_k} x_i$ of monomials (where the index sets $I_1, \ldots, I_k$ are disjoint and $n = \sum_{j=1}^{k} |I_j|$) satisfies $2^n - 2w_H(f) = \prod_{j=1}^{k}(2^{|I_j|} - 2)$. The Hamming weight of the function whose support equals all vectors of a Hamming weight of at least k equals $\sum_{i=k}^{n}\binom{n}{i}$. We find in both cases rather peculiar weights and, in the latter case, the algebraic degree needs to be determined.

There exist also secondary constructions of Boolean functions:

- The direct sum, already recalled above in the particular context of monomials, consists of adding functions whose sets of variables are disjoint. It gives weights that are a little peculiar: We have recalled above that if f is the direct sum of a t-variable function f_1 and a $(n-t)$-variable function f_2, then the character sum of f equals the product of the character sums of f_1 and f_2, and this implies:

$$2^n - 2w_H(f) = (2^t - 2w_H(f_1))(2^{n-t} - 2w_H(f_2)).$$

  This construction is interesting because it does not need particular precautions about the algebraic degree of f, which equals the maximum of the algebraic degrees of f_1 and f_2. Hence, for every weight w_1 in $RM(r, t)$ and every weight w_2 in $RM(r, n-t)$, the number w such that $2^n - 2w = (2^t - 2w_1)(2^{n-t} - 2w_2)$ is a weight in $RM(r, n)$), with the convention that if $r > t$, then $RM(r, t)$ equals

$RM(t,t)$ (and can then have the weight of any integer between 0 and 2^t). With this construction, there is a systematic way of building weights in $RM(r,n)$ from weights in $RM(r,t)$ and $RM(r,n-t)$.

- The indirect sum (see [8, Sections 6.1.16 and 7.1.9]) also deals with functions whose sets of variables are disjoint, but in a more complex way: We have two functions f_1, f_2 on the same set of t variables, two functions g_1 and g_2 on the same set of $n-t$ variables, disjoint from the previous one, and $f(x,y) = f_1(x) + g_1(x) + (f_1(x) + f_2(x))(g_1(x) + g_2(x))$. We then have $\sum_{x\in\mathbb{F}_2^t, y\in\mathbb{F}_2^{n-t}}(-1)^{f(x,y)} = \frac{1}{2}(\sum_{x\in\mathbb{F}_2^t}(-1)^{f_1(x)})[\sum_{y\in\mathbb{F}_2^{n-t}}(-1)^{g_1(y)} + \sum_{y\in\mathbb{F}_2^{n-t}}(-1)^{g_2(y)}] + \frac{1}{2}(\sum_{x\in\mathbb{F}_2^t}(-1)^{f_2(x)})[\sum_{y\in\mathbb{F}_2^{n-t}}(-1)^{g_1(y)} - \sum_{y\in\mathbb{F}_2^{n-t}}(-1)^{g_2(y)}]$ and, therefore:

$$2^n - 2w_H(f) =$$

$$(2^t - 2w_H(f_1))\left[2^{n-t} - w_H(g_1) - w_H(g_2)\right] + (2^t - 2w_H(f_2))\left[w_H(g_2) - w_H(g_1)\right].$$

  The algebraic degree of f is not automatically bounded by r from above, unless we take the initial functions f_1, f_2 in $RM(s,t)$ with $s \le r$ and the initial functions g_1, g_2 in $RM(r-s,n-t)$ but this does not allow to provide interesting weights. If we take f_1, f_2 in $RM(r,t)$ and g_1, g_2 in $RM(r,n-t)$, this construction provides weights that are possibly less peculiar than with the direct sum, but in a much less systematic way, because we need to take care of the algebraic degree.
- The sum without extension of the number of variables (see [8, Sections 6.1.16 and 7.1.9]) takes three n-variable Boolean functions f_1, f_2, f_3 and defines the Boolean function $f = f_1f_2+f_1f_3+f_2f_3$. We have:

$$w_H(f) = \frac{1}{2}\Big(w_H(f_1) + w_H(f_2) + w_H(f_3) - w_H(f_1 + f_2 + f_3)\Big).$$

  This secondary construction has been introduced because of the nice behavior of its Walsh transform, but it has the same drawback as the indirect sum about the algebraic degree of f.
- The so-called $(u|u+v)$-construction (see [22]) allows us to construct all of $RM(r,n)$ from $RM(r-1,n-1)$ and $RM(r,n-1)$. It corresponds to the fact that an n-variable Boolean function $f(x_1,\ldots,x_n)$ can be written in the form $f_0(x_1,\ldots,x_{n-1}) + x_nf_1(x_1,\ldots,x_{n-1})$ and has an algebraic degree of at most r if and only if f_0 has an algebraic degree of at most r and f_1 has an algebraic degree of at most $r-1$. The corresponding codeword is the concatenation of the codewords in $RM(r,n-1)$ associated to f_0 and $f_0 + f_1$, and for the Hamming weight, it has the sum of the Hamming weights of these two functions.
  The pairs $(f_0, f_0 + f_1)$, when f_0 ranges over $RM(r,n-1)$ and f_1 ranges over $RM(r-1,n-1)$, do not provide all possible pairs of codewords in $RM(r,n-1)$ because of the restriction that f_1 has an algebraic degree of at most $r-1$, but if we impose that f_0 itself ranges over $RM(r-1,n-1)$, then the weights of the resulting codewords of $RM(r,n)$ range over the sums of two weights in $RM(r-1,n-1)$. This leads to a result given in [30] and used in [12]: For all pairs of integers (r,n) with $0 \le r \le n$, the weight spectrum of $RM(r,n)$ includes as a subset $S+S$, where S is the weight spectrum of $RM(r-1,n-1)$. This result has allowed us to obtain the weight spectra of infinite classes of Reed-Muller codes, but only for orders that are very close to n.

A completely different way of evaluating weights in Reed-Muller codes consists of the fact that, for every Boolean function f of an algebraic degree of at most r, we have $\Big(\sum_{x\in\mathbb{F}_2^n}(-1)^{f(x)}\Big)^2 = \sum_{a\in\mathbb{F}_2^n}\sum_{x\in\mathbb{F}_2^n}(-1)^{D_af(x)}$, where for every a, the so-called derivative $D_af(x) = f(x) + f(x+a)$ has an algebraic degree of at most $r-1$. If we are able to determine the weights of all these derivatives, we

obtain the absolute value of $\sum_{x\in\mathbb{F}_2^n}(-1)^{f(x)} = 2^n - 2w_H(f)$, and since every Reed-Muller code is invariant under the complementation of its codewords, this provides two weights if $\sum_{x\in\mathbb{F}_2^n}(-1)^{f(x)} \neq 0$. However, this method, which is clearly more efficient for low orders r, is better suited for determining some specific weights than for systematically finding new weights in infinite classes of Reed-Muller codes.

It is then useful to find a new way, as systematic as possible, for providing weights (hopefully previously unknown) and codewords having such weights.

3. A new construction of Boolean functions with an algebraic degree bounded from above

In this section, we present our construction. It comes from a formula that is satisfied by all Boolean functions of an algebraic degree bounded from above by some number s (and therefore by all vectorial functions $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ of an algebraic degree of at most s). This formula has been originally found and used (in [11]) in the framework of countermeasures against side channel attacks, a domain of applied cryptography. It also corresponds to what we call zero-sum sets, a notion used in the cryptanalysis of block ciphers. It could seem rather unrelated to coding theory in general and to the determination of weights in Reed-Muller codes in particular, but it is not, as we shall see. This formula depends on parameters (that are elements of $\mathbb{F}_2^n$) and will lead to numerous Boolean functions f of the algebraic degree bounded from above, since the Hamming weight of these functions can be determined, to numerous weights in Reed-Muller codes.

3.1. Degree-s zero-sum sets as Reed-Muller codewords

A set $S \subseteq \mathbb{F}_2^n$ is called degree-s zero-sum[‖] if we have $\sum_{x\in S} f(x) = 0$ for every n-variable Boolean function f of an algebraic degree of at most s (and then $\sum_{x\in S} F(x) = 0$ for every vectorial function F in n variables of an algebraic degree of at most s).

The degree-s zero-sum sets are then the supports of the codewords in the dual code of $RM(s, n)$. The dual of $RM(s, n)$ equals $RM(r, n)$ where $r = n - s - 1$ [22] and degree-s zero-sum sets are then the supports of the n-variable Boolean functions of an algebraic degree of at most r, that is, of the codewords of $RM(r, n)$. Hence, determining the possible sizes of degree-s zero-sum sets is directly related to determining the weights in Reed-Muller codes.

3.2. A construction of Boolean functions with bounded algebraic degree

We know from [11, Corollary 1] that if an n-variable Boolean function $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ or, more generally a vectorial (n, m)-function $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$, has an algebraic degree of at most s, then for every $t > s$ and for every $a_1, \ldots, a_t \in \mathbb{F}_2^n$, we have

$$F\Big(\sum_{i=1}^{t} a_i\Big) = \sum_{j=0}^{s} \mu_{t,s}(j) \sum_{J\subseteq\{1,\ldots,t\};|J|=j} F\Big(\sum_{i\in J} a_i\Big) \tag{3.1}$$

(the sum, of course, being calculated modulo 2) where

$$\mu_{t,s}(j) = \binom{t-j-1}{s-j} \bmod 2$$

[‖]This term comes from [3], which denotes by d what we denote here by s; we prefer using s to avoid any confusion with the minimum distance of codes.

for every $j \leq s$, with the conventions $\binom{l}{0} = 1$ for every l and $\sum_{i\in\emptyset} a_i = 0$.

According to (3.1), the set of all the elements a of $\mathbb{F}_2^n$, which appear an odd number of times as $a = \sum_{i=1}^{t} a_i$, or $a = \sum_{i\in J} a_i$ where J has size at most s and $\mu_{t,s}(|J|) = 1$, is a degree-s zero-sum set. We then have the following result, in which, for every $a \in \mathbb{F}_2^n$, we denote by δ_a the Boolean function over $\mathbb{F}_2^n$ which takes value 1 at a and 0 everywhere else (such a function can be called an atomic, or Dirac, or Kronecker function):

Theorem 1. *Let n, $s \geq 0$ and $t \geq 1$ be integers such that $s < t$ and $s < n$. Given any elements $a_1, \ldots, a_t$ of $\mathbb{F}_2^n$, the Boolean function:*

$$f^{(s)}_{a_1,\ldots,a_t} := \delta_{\sum_{i=1}^{t} a_i} + \sum_{j=0}^{s} \mu_{t,s}(j) \sum_{J\subseteq\{1,\ldots,t\};|J|=j} \delta_{\sum_{i\in J} a_i}, \tag{3.2}$$

(where $\mu_{t,s}(j) = \binom{t-j-1}{s-j} \bmod 2 = \binom{t-j-1}{t-s-1} \bmod 2$), has an algebraic degree of at most $r = n - s - 1$.

Remark 1. *(1) $f^{(s)}_{a_1,\ldots,a_t}$ is in general not a symmetric function (that is, its value changes when we permute its input bits) despite the fact that its expression (3.2) is symmetric with respect to $a_1, \ldots, a_t$ (*i.e.*, its value does not change when we permute the a_i's).*
(2) For every positive integers n, s, t such that $s < n$ and $s < t$, and every $a_1, \ldots, a_t$ in $\mathbb{F}_2^n$, all the functions $f^{(s)}_{a_1,\ldots,a_t}, f^{(s+1)}_{a_1,\ldots,a_t}, \ldots, f^{(t-1)}_{a_1,\ldots,a_t}$ have algebraic degrees of at most r.
(3) Suppose that for some n, s, t, the function $f^{(s)}_{a_1,\ldots,a_t}$ has an algebraic degree $r' < n - s - 1$, then it is orthogonal to every codeword of the Reed-Muller code $RM(n - r' - 1, n)$ with $n - r' - 1 > s$, and it is, therefore, orthogonal to the Reed-Muller code $RM(s + 1, n)$, whose elements satisfy the Relation (3.1). There seems to most often exist codewords of $RM(s + 1, n)$ which do not satisfy Relation (3.1). We deduce that, most often, $f^{(s)}_{a_1,\ldots,a_t}$ has in fact an algebraic degree of $r = n-s-1$ exactly. Examples 1 and 2 will illustrate this, but there are also examples where $f^{(s)}_{a_1,\ldots,a_t}$ has an algebraic degree strictly smaller; see, for instance, Proposition 1.

Example 1. *(toy example) Let $n = 3, s = 1$ (and, therefore, $r = 1$), $t = 4, a_1 = (1, 0, 0), a_2 = (0, 1, 0), a_3 = (0, 0, 1)$, and $a_4 = (1, 1, 1)$. We have $\mu_{t,s}(0) = \binom{3}{1} \pmod 2 = 1, \mu_{t,s}(1) = \binom{2}{0} \pmod 2 = 1$. Hence, $f^{(s)}_{a_1,a_2,a_3,a_4} = \delta_{a_1+a_2+a_3+a_4} + \delta_{(0,0,0)} + \delta_{a_1} + \delta_{a_2} + \delta_{a_3} + \delta_{a_4}$ is the indicator function of the affine plane $\{a_1, a_2, a_3, a_4\}$.*

3.2.1. Linear equivalence between the constructed functions when $a_1, \ldots, a_t$ are linearly independent

We say that two n-variable Boolean functions f, g are linearly (resp., affinely) equivalent if there exists a linear automorphism (resp., an affine automorphism) L of $\mathbb{F}_2^n$ such that $g = f \circ L$, then f and g have the same Hamming weight and the same algebraic degree. All the functions in a same equivalence class contribute then for the same weight in the weight spectrum of the corresponding Reed-Muller code. We are then interested, when we find a function with a known algebraic degree and weight, to know whether it is inequivalent to previously found functions. For $t \leq n$, two choices "$a_1, \ldots, a_t$", respectively, "$a'_1, \ldots, a'_t$", of linearly independent elements give linearly equivalent functions $f^{(s)}_{a_1,\ldots,a_t}$ and $f^{(s)}_{a'_1,\ldots,a'_t}$, because there exists a linear automorphism L, mapping $a_1, \ldots, a_t$ to $a'_1, \ldots, a'_t$, respectively, and, therefore, mapping $\sum_{i\in J} a_i$ to $\sum_{i\in J} a'_i$ for every J. We then have $f^{(s)}_{a_1,\ldots,a_t} = f^{(s)}_{a'_1,\ldots,a'_t} \circ L$.

3.2.2. Studying some particular cases of (t, s) when $a_1, \ldots, a_t$ are not necessarily linearly independent

For two choices $a_1, \ldots, a_t$ and $a'_1, \ldots, a'_t$, of linearly dependent elements, the corresponding functions $f^{(s)}_{a_1,\ldots,a_t}$ and $f^{(s)}_{a'_1,\ldots,a'_t}$ may not be affine equivalent. Of course, if $a_1, \ldots, a_t$ and $a'_1, \ldots, a'_t$ satisfy exactly the same linear relations over $\mathbb{F}_2$, then there is again a linear automorphism mapping $a_1, \ldots, a_t$ to $a'_1, \ldots, a'_t$, respectively (indeed, the two families have the same rank k; we can choose in each family k elements generating the other elements of the family by the same relations and deduce such linear automorphism), but if not, then the functions $f^{(s)}_{a_1,\ldots,a_t}$ and $f^{(s)}_{a'_1,\ldots,a'_t}$ may be inequivalent.

Before seeing an example where $f^{(s)}_{a_1,\ldots,a_t}$ and $f^{(s)}_{a'_1,\ldots,a'_t}$ are not affine equivalent, let us systematically visit the first possible values of s (for any $t > s$):

• *Case* $s = 0$: For $t \geq 1$, we have $f^{(0)}_{a_1,\ldots,a_t} = \delta_{\sum_{i=1}^t a_i} + \delta_0$, which can have a weight of either 0 or 2; we get then only the two smallest weights of $RM(n, n-1)$;

• *Case* $s = 1$: For $t \geq 2$, we have $f^{(1)}_{a_1,\ldots,a_t} = \delta_{\sum_{i=1}^t a_i} + (t-1)\,\delta_0 + \sum_{i=1}^t \delta_{a_i}$ (we omit the "mod2"); if t is even, then we get $\delta_{\sum_{i=1}^t a_i} + \delta_0 + \sum_{i=1}^t \delta_{a_i}$, which has an even weight of at most $t+2$, and if n is odd, then we get $\delta_{\sum_{i=1}^t a_i} + \sum_{i=1}^t \delta_{a_i}$, which has an even weight as well of at most $t+1$; Since t is not bounded above, we get all possible weights of $RM(n, n-2)$ (and this case is then very different from the previous one): We can easily check that the weights 2 and $2^n - 2$ are impossible and all other even weights between 0 and 2^n are possible; for instance, weight 4 is achieved by taking either $t = 2$ and a_1, a_2 nonzero and distinct (i.e., linearly independent over $\mathbb{F}_2$) or $t = 3$ and a_1, a_2, a_3 distinct;

• *Case* $s = 2$: For $t \geq 3$, we have $f^{(2)}_{a_1,\ldots,a_t} = \delta_{\sum_{i=1}^t a_i} + \binom{t-1}{2}\delta_0 + (t-2)\sum_{i=1}^t \delta_{a_i} + \sum_{1\leq i<j\leq t} \delta_{a_i+a_j}$; hence, if all the sums $a_i + a_j$ and the a_i are nonzero and distinct, we have a function of a Hamming weight in $[\![\binom{t}{2} + t - 1, \binom{t}{2} + t + 2]\!]$ if t is odd and in $[\![\binom{t}{2} - 1, \binom{t}{2} + 2]\!]$ if t is even. If we only assume that all the sums $a_i + a_j$ are distinct, we have a function of a Hamming weight of at least $\binom{t}{2} - 2 - t = \frac{t^2-3t-4}{2}$ if t is odd and $\binom{t}{2} - 2 = \frac{t^2-t-4}{2}$ if t is even.

• *Case* $s = 3$: For $t \geq 4$, we have $f^{(3)}_{a_1,\ldots,a_t} = \delta_{\sum_{i=1}^t a_i} + \binom{t-1}{3}\delta_0 + \binom{t-2}{2}\sum_{i=1}^t \delta_{a_i} + (t-3)\sum_{1\leq i<j\leq t} \delta_{a_i+a_j} + \sum_{1\leq i<j<k\leq t} \delta_{a_i+a_j+a_k}$; hence, if all the sums $a_i + a_j + a_k$ are distinct, we have a function of a Hamming weight of at least $\binom{t}{3} - 2 - t - \binom{t}{2} = \frac{t^3-6t^2-t-12}{6}$ if t is even and $\binom{t}{3} - 2 - t = \frac{t^3-3t^2-4t-12}{6}$ if t is odd.

Since, for the same value of n and the same value of t, $f^{(1)}_{a_1,\ldots,a_t} = \delta_{\sum_{i=1}^t a_i} + (t-1)\,\delta_0 + \sum_{i=1}^t \delta_{a_i}$ can have different Hamming weights according to the values of the a_i's when they are linearly dependent, we have an example where $f^{(s)}_{a_1,\ldots,a_t}$ and $f^{(s)}_{a'_1,\ldots,a'_t}$ are not affine equivalent, even if $a_1, \ldots, a_t$ are distinct as well as $a'_1, \ldots, a'_t$.

Let us now systematically visit the first possible values of $t > s$ (for any s):

• *For* $t = s + 1$, we have $\mu_{t,s}(j) = \binom{s-j}{s-j} \bmod 2 = 1$ for all $j \leq s$. Note that this was expected since Relation (3.1) expresses, in particular, that for a function of degree of at most s, the sum of the values of the function taken over any $(s+1)$-dimensional affine space equals 0. The Hamming weight $w_{s+1,s}$ of $f^{(s)}_{a_1,\ldots,a_{s+1}}$ is at most $1 + \sum_{j=0}^s \binom{t}{j} = 2^{s+1}$. Hence, since 2^{s+1} equals the minimum distance of $RM(r, n)$, the Hamming weight of $f^{(s)}_{a_1,\ldots,a_{s+1}}$ is either zero or 2^{s+1} (depending on the choice of $a_1, \ldots, a_{s+1}$). More precisely:

Proposition 1. *For every* $s \geq 0$ *and every linearly independent* $a_1, \ldots, a_{s+1}$ *in* $\mathbb{F}_2^n$, $f^{(s)}_{a_1,\ldots,a_{s+1}}$ *is the minimum weight codeword in* $RM(r, n)$ *whose support equals* $\langle a_1, \ldots, a_{s+1}\rangle$, *the vector space over* $\mathbb{F}_2$ *generated by* $a_1, \ldots, a_{s+1}$. *If* $a_1, \ldots, a_{s+1}$ *are linearly dependent, then* $f^{(s)}_{a_1,\ldots,a_{s+1}}$ *equals the zero function.*

Proof. We have $f^{(s)}_{a_1,\dots,a_{s+1}} = \sum_{J\subseteq\{1,\dots,s+1\}} \delta_{\sum_{i\in J} a_i}$. If $a_1,\dots,a_{s+1}$ are linearly independent, then $f^{(s)}_{a_1,\dots,a_{s+1}}$ equals the indicator of the vector space generated by $a_1,\dots,a_{s+1}$ (and we obtain with the functions $f^{(s)}_{a_1,\dots,a_t}$ all the minimum weight codewords in $RM(r,n)$). If $a_1,\dots,a_{s+1}$ are linearly dependent, then the Hamming weight of $f^{(s)}_{a_1,\dots,a_t}$ is strictly less than the minimum distance of $RM(r,n)$, and it is then 0. Note that, assuming (without loss of generality, thanks to the invariance of $f^{(s)}_{a_1,\dots,a_t}$ when permuting the a_i's) that $a_t = a_1 + \cdots + a_k$, for some $k < t$, it is easily seen that each Dirac function obtained after replacing a_t by its value in the expression of $f^{(s)}_{a_1,\dots,a_{s+1}}$ appears an even number of times. This implies that this expression cancels. □

• *For* $t = s+2$, we have $\mu_{t,s}(j) = \binom{s+1-j}{s-j} \bmod 2 = (s+1-j) \bmod 2$ and $f^{(s)}_{a_1,\dots,a_t} = \delta_{\sum_{i=1}^t a_i} + \sum_{k=0}^{\lfloor \frac{s}{2} \rfloor} \sum_{\substack{J\subseteq\{1,\dots,t\}\\ |J|=s-2k}} \delta_{\sum_{i\in J} a_i}$.

We have $w_{s+2,s} \le 1 + \sum_{k=0}^{\lfloor \frac{s}{2} \rfloor} \binom{s+2}{s-2k} = 1 + \sum_{k=0}^{\lfloor \frac{s}{2} \rfloor} \binom{s+2}{2k+2} = 2^{s+1}$. More precisely:

Proposition 2. *For every $s \ge 0$ and every linearly independent $a_1,\dots,a_{s+2}$ in $\mathbb{F}_2^n$, $f^{(s)}_{a_1,\dots,a_{s+2}}$ is a minimum weight codeword in $RM(r,n)$. If $a_1,\dots,a_{s+1}$ are linearly dependent, then $f^{(s)}_{a_1,\dots,a_{s+1}}$ can equal a minimum weight codeword in $RM(r,n)$ or the zero function.*

• *For* $t = s+3$, we have that $\mu_{t,s}(j) = \binom{s+2-j}{s-j} \bmod 2 = \binom{s+2-j}{2} \bmod 2$ equals $\begin{cases} 1 \text{ if } s+2-j \bmod 4 \in \{2,3\} \\ 0 \text{ if } s+2-j \bmod 4 \in \{0,1\} \end{cases}$, and we have:

$$f^{(s)}_{a_1,\dots,a_t} = \delta_{\sum_{i=1}^t a_i} + \sum_{\substack{0\le j\le s\\ s+2-j\equiv 2,3 \bmod 4}} \sum_{\substack{J\subseteq\{1,\dots,t\}\\ |J|=j}} \delta_{\sum_{i\in J} a_i}.$$

3.3. On the weights of the constructed functions

The interest of Theorem 1 is that it is possible to calculate mathematically, under some conditions, the Hamming weight of $f^{(s)}_{a_1,\dots,a_t}$, and that the weights obtained do not look peculiar.

Proposition 3. *Let n, $s \ge 0$ and $t \ge 1$ be integers such that $s < t$ and $s < n$. For any elements $a_1,\dots,a_t$ of $\mathbb{F}_2^n$, let $f^{(s)}_{a_1,\dots,a_t}$ be the Boolean function given by (3.2). If $a_1,\dots,a_t$ are linearly independent over $\mathbb{F}_2$, then $f^{(s)}_{a_1,\dots,a_t}$ has Hamming weight:*

$$w_{t,s} = 1 + \sum_{\substack{j\in\{0,\dots,s\};\\ \mu_{t,s}(j)=1}} \binom{t}{j}, \tag{3.3}$$

where $\mu_{t,s}(j) = \binom{t-j-1}{s-j} \bmod 2$, and otherwise, it has a Hamming weight of at most $w_{t,s}$.

Indeed, the former assertion comes from the fact that, for any two distinct J, the corresponding elements $\sum_{i\in J} a_i$ are distinct, since $a_1,\dots,a_t$ are linearly independent over $\mathbb{F}_2$, and the latter is obvious. Note that the Hamming weight of $f^{(s)}_{a_1,\dots,a_t}$ has necessarily the same parity as $w_{t,s}$ since the atomic functions involved in (3.2) cancel by pairs, but since we already know that this weight is even because $r = n-s-1$ is strictly smaller than n, this only tells us that $w_{t,s}$ is even (while it may not always be a weight in $RM(r,n)$ when $t > n$). Note also that $w_{t,s} \ge 1 + \binom{t}{s}$ since $\mu_{t,s}(s) = 1$ (and then the weight of $f^{(s)}_{a_1,\dots,a_t}$ cannot equal $w_{t,s}$ if $\binom{t}{s} \ge 2^n$), and that if $t-s$ is odd, then $w_{t,s} \ge 1 + \binom{t}{s-1} + \binom{t}{s}$, since $\mu_{t,s}(s-1) = t-s$ (and then the weight of $f^{(s)}_{a_1,\dots,a_t}$ cannot equal $w_{t,s}$ if $\binom{t}{s-1} + \binom{t}{s} \ge 2^n$).

Example 2. *Let us take $n = 12$, $r = 8$. We can check that $f^{(s)}_{a_1,\dots,a_t}$ can reach weight 166 in two cases where $a_1, \dots, a_t$ are linearly independent over $\mathbb{F}_2$. Indeed, for having $r = n - s - 1 = 8$, we need to take $s = 3$. For the weight $w_{t,s} = 1 + \sum_{\substack{j\in\{0,\dots,s\};\\ \mu_{t,s}(j)=1}} \binom{t}{j}$ given by Proposition 3 to equal 166, we need to take $t \in \{10, 11\}$. Recall that all these functions are affine equivalent, for a fixed value of t. Denoting by $(e_1, \dots, e_{12})$ the canonical basis of $\mathbb{F}_2^{12}$ (made of all weight 1 vectors), we obtain then two classes of functions, that are respectively affine equivalent to $f^{(3)}_{e_1,\dots,e_{10}} = \delta_{\sum_{i=1}^{10} e_i} + \sum_{\substack{0\le j\le 3\\ \binom{9-j}{3-j} \bmod 2=1}} \sum_{J\subseteq\{1,\dots,10\};|J|=j} \delta_{\sum_{i\in J} e_i} = \delta_{\sum_{i=1}^{10} e_i} + \sum_{J\subseteq\{1,\dots,10\};|J|=2} \delta_{\sum_{i\in J} e_i} + \sum_{J\subseteq\{1,\dots,10\};|J|=3} \delta_{\sum_{i\in J} e_i}$ and $f^{(3)}_{e_1,\dots,e_{11}} = \delta_{\sum_{i=1}^{11} e_i} + \sum_{\substack{0\le j\le 3\\ \binom{10-j}{3-j} \bmod 2}} \sum_{J\subseteq\{1,\dots,11\};|J|=j} \delta_{\sum_{i\in J} e_i} = \delta_{\sum_{i=1}^{11} e_i} + \sum_{J\subseteq\{1,\dots,11\};|J|=3} \delta_{\sum_{i\in J} e_i}$.*

Recall (see e.g., *[8, Subsection 10.1.1]) that denoting by $1_{E_{n,j}}$ the n-variable Boolean function whose support is the set $E_{n,j}$ of all vectors of a Hamming weight of j in $\mathbb{F}_2^n$, we have $1_{E_{n,j}}(x) = \sum_{\substack{I\subseteq\{1,\dots,n\}\\ \binom{|I|}{j}=1 \pmod 2}} x^I$.*

We then have $f^{(3)}_{e_1,\dots,e_{10}}(x) = (x_{11}+1)(x_{12}+1)\Big[\prod_{i=1}^{10} x_j + \sum_{\substack{I\subseteq\{1,\dots,10\}\\ |I|\in\{2,3,6,7,10\}}} \prod_{i\in I} x_i + \sum_{\substack{I\subseteq\{1,\dots,10\}\\ |I|\in\{3,7\}}} \prod_{i\in I} x_i\Big]$ and $f^{(3)}_{e_1,\dots,e_{11}}(x) = (x_{12}+1)\Big[\prod_{i=1}^{11} x_j + \sum_{\substack{I\subseteq\{1,\dots,11\}\\ |I|\in\{3,7,11\}}} \prod_{i\in I} x_i\Big]$ and these two functions both have an algebraic degree of 8.

It is interesting to notice that $w_{t,s}$, defined in Relation (3.3), does not depend on n (we only have the condition that $n \ge t$). Of course, n plays a role through the value of r.

We can see that the weights provided by Proposition 3 are few for low orders (since t ranges from $n - r$ to n) and a little more numerous for large orders.

We now observe a property of $w_{t,s}$ that seems easier to show by considering Relation (3.3) than to infer directly from the way $f^{(s)}_{a_1,\dots,a_t}$ was derived:

Lemma 1. *For every $s, i \ge 0$, we have $w_{s+2i+1,s} = w_{s+2i+2,s} \le w_{s+2i+3,s}$ and this latter inequality is strict for $s > 0$.*

Proof. According to Lucas' theorem (see, e.g., [22, Page 404]), we have that $\mu_{s+2i+1,s}(s-j) = \binom{2i+j}{j} \bmod 2$ equals 1 if, and only if, the binary expansion of j is covered by that of $2i + j$, then $\mu_{s+2i+1,s}(s - j)$ has the same value for $j = 2k$ and $j = 2k + 1$, while $\mu_{s+2i+2,s}(s - j) = \binom{2i+j+1}{j} \bmod 2$ shares the same value if $j = 2k$ and equals 0 if $j = 2k + 1$. We deduce that $w_{s+2i+2,s} = 1 + \sum_{\substack{0\le j\le s;j\,even;\\ \mu_{s+2i+1,s}(s-j)=1}} \binom{s+2i+2}{s-j} = 1 + \sum_{\substack{0\le j\le s;j\,even;\\ \mu_{s+2i+1,s}(s-j)=1}} \left(\binom{s+2i+1}{s-j} + \binom{s+2i+1}{s-j-1}\right) = 1 + \sum_{\substack{j\ge 0;j\,even;\\ \mu_{s+2i+1,s}(s-j)=1}} \binom{s+2i+1}{s-j} + \sum_{\substack{j\ge 0;j\,odd;\\ \mu_{s+2i+1,s}(s-j)=1}} \binom{s+2i+1}{s-j} = w_{s+2i+1,s}$. This proves the equality.

We have $\mu_{s+2i+3,s}(s-j) = \binom{2i+2+j}{j} \bmod 2 = \left(\binom{2i+1+j}{j} + \binom{2i+1+j}{j-1}\right) \bmod 2 = \mu_{s+2i+2,s}(s-j) + \mu_{s+2i+3,s}(s-j+1)$. If $\mu_{s+2i+3,s}(s - j + 1) = 0$, then $\mu_{s+2i+3,s}(s - j) = \mu_{s+2i+2,s}(s - j)$. Using for $\mu_{s+2i+3,s}(s - j + 1) = 0$ that $\binom{s+2i+3}{s-j}$ is strictly larger than $\binom{s+2i+2}{s-j}$ and for $\mu_{s+2i+3,s}(s-j+1) = 1$ that $\binom{s+2i+3}{s-j+1}$ equals $\binom{s+2i+2}{s-j} + \binom{s+2i+2}{s-j+1}$, we deduce that $w_{s+2i+3,s} = 1 + \sum_{\substack{j\in\{0,\dots,s\};\\ \mu_{s+2i+3,s}(s-j)=1}} \binom{s+2i+3}{s-j} \ge w_{s+2i+2,s} = 1 + \sum_{\substack{j\in\{0,\dots,s\};\\ \mu_{s+2i+2,s}(s-j)=1}} \binom{s+2i+2}{s-j}$ and the inequality is then verified. Moreover, if $s > 0$, then the inequality is strict. □

Open problem: Find a direct explanation of Lemma 1, from the proofs of [11, Theorem 1 and Corollary 1].

Remark 2. *We have seen in the proof of Lemma 1 that $\mu_{s+2i+2,s}(s-j)$ equals 0 for every odd j. Hence, if all the elements $\sum_{i\in J} a_i$ are distinct for all $J \subset \{1,\dots,s+2i+2\}$ whose sizes are at most s and have the same parity as s, then $f^{(s)}_{a_1,\dots,a_{s+2i+2}}$ has a Hamming weight of $w_{s+2i+2,s}$ as well. This is possible with $s+2i+2 > n$: Take, for instance, $s=2$, $i=1$, $n=s+2i+1=5$, $t=6$, and $a_6=a_5+a_4$, then $\mu(t,s)(0)=0, \mu_{t,s}(1)=0, \mu_{t,s}(2)=1$, and $a_1+\cdots+a_6 = a_1+a_2+a_3, a_1+a_2, a_1+a_3, a_1+a_4, a_12+a_5, a_1+a_6 = a_1+a_4+a_5, a_2+a_3, a_2+a_4, a_2+a_5, a_2+a_6 = a_2+a_4+a_5, a_3+a_4, a_3+a_5, a_3+a_6 = a_3+a_4+a_5, a_4+a_5, a_4+a_6 = a_5, a_5+a_6 = a_4$ are all distinct.*

Open problem: Determine the exact Hamming weight of $f^{(s)}_{a_1,\dots,a_t}$ by means of s, n and $a_1,\dots,a_t$ when the latter are linearly dependent over $\mathbb{F}_2$.

In the next corollary we call the weight spectrum of $RM(r,n)$ the list of all possible weights in $RM(r,n)$)

Corollary 1. *Whatever the positive integers of n and $r<n$ are, the weight spectrum of $RM(r,n)$ contains all the numbers:*

$$\begin{aligned} w_{t,n-r-1} &\;;\; t=n-r,\dots,n;\\ w_{t,n-r} &\;;\; t=n-r+1,\dots,n;\\ &\vdots\\ w_{t,n-2} &\;;\; t=n-1,n. \end{aligned}$$

Indeed, for every $t \le n$, there exist t linearly independent elements.

In Table 1, we give for $n \le 21$ and for all $r=1,\dots,n-1$, the list in regular roman** of the values $w_{t,n-r-1}$ where t ranges from $n-r$ to n. All these numbers are weights in $RM(r,n)$, and all the lists displayed for the input pairs $(n,r),(n,r-1),\dots,(n,1)$ provide weights in $RM(r,n)$. We can check on these lists that Lemma 1 is verified, that is, the numbers go by pairs of consecutive equal values and the lists are nondecreasing.

We can find in Table 1 many numbers which were not known before as weights in $RM(r,n)$, such as 3004 or 6436 in $RM(6,14)$.

**The values in bold will be obtained below in Subsection 3.3.1.

Table 1. Lists of values of $w_{n-r,n-r-1}, \ldots, w_{n,n-r-1}; \mathbf{w'_{t,n-r-1}}$.

n	r	$[w_{n-r,n-r-1}, \ldots, w_{n,n-r-1}; \mathbf{w'_{t,n-r-1}}]$
3		
	1	[4 4]
	2	[2 2 2]
4		
	1	[8 8]
	2	[4 4 6]
	3	[2 2 2 2]
5		
	1	[16 16]
	2	[8 8 16]
	3	[4 4 6 6 **8]**
	4	[2 2 2 2 2]
6		
	1	[32 32]
	2	[16 16 36]
	3	[8 8 16 16 **24]**
	4	[4 4 6 6 8]
	5	[2 2 2 2 2 2]
7		
	1	[64 64]
	2	[32 32 72]
	3	[16 16 36 36 **56]**
	4	[8 8 16 16 30 **24**]
	5	[4 4 6 6 8 8 **10]**
	6	[2 2 2 2 2 2 2]
8		
	1	[128 128]
	2	[64 64 136]
	3	[32 32 72 72 **112]**
	4	[16 16 36 36 94 **56**]
	5	[8 8 16 16 30 30 **24 40**]
	6	[4 4 6 6 8 8 10]
	7	[2 2 2 2 2 2 2 2]
9		
	1	[256 256]
	2	[128 128 256]
	3	[64 64 136 136 **208]**
	4	[32 32 72 72 256 **112**]
	5	[16 16 36 36 94 94 **56 124]**
	6	[8 8 16 16 30 30 46 **24 40]**
	7	[4 4 6 6 8 8 10 10 **12]**
	8	[2 2 2 2 2 2 2 2 2]

n	r	$[w_{n-r,n-r-1}, \ldots, w_{n,n-r-1}; \mathbf{w'_{t,n-r-1}}]$
10		
	1	[512 512]
	2	[256 256 496]
	3	[128 128 256 256 **384]**
	4	[64 64 136 136 628 **208]**
	5	[32 32 72 72 256 256 **112 328]**
	6	[16 16 36 36 94 94 166 **56 124]**
	7	[8 8 16 16 30 30 46 46 **24 40 62]**
	8	[4 4 6 6 8 8 10 10 12 **14]**
	9	[2 2 2 2 2 2 2 2 2]
11		
	1	[1024 1024]
	2	[512 512 992]
	3	[256 256 496 496 **736]**
	4	[128 128 256 256 1420 **384]**
	5	[64 64 136 136 628 628 **208 784]**
	6	[32 32 72 72 256 256 496 **112 328]**
	7	[16 16 36 36 94 94 166 166 **56 124 238]**
	8	[8 8 16 16 30 30 46 46 68 **40 62]**
	9	[4 4 6 6 8 8 10 10 12 12]
	10	[2 2 2 2 2 2 2 2 2 2]
12		
	1	[2048 2048]
	2	[1024 1024 2016]
	3	[512 512 992 992 **1472]**
	4	[256 256 496 496 3004 **736]**
	5	[128 128 256 256 1420 1420 **384 1744]**
	6	[64 64 136 136 628 628 1288 **208 784]**
	7	[32 32 72 72 256 256 496 496 **112 328 736]**
	8	[16 16 36 36 94 94 166 166 300 **124 238 300]**
	9	[8 8 16 16 30 30 46 46 68 68 **24 40 62 86]**
	10	[4 4 6 6 8 8 10 10 12 12 14]
	11	[2 2 2 2 2 2 2 2 2 2 2]

n	r	$[w_{n-r,n-r-1}, \ldots, w_{n,n-r-1}; \mathbf{w'_{t,n-r-1}}]$
13		
	1	[4096 4096]
	2	[2048 2048 4096]
	3	[1024 1024 2016 2016 **3008]**
	4	[512 512 992 992 6008 **1472]**
	5	[256 256 496 496 3004 3004 **736 3664]**
	6	[128 128 256 256 1420 1420 3004 **384 1744]**
	7	[64 64 136 136 628 628 1288 1288 **208 784 1948]**
	8	[32 32 72 72 256 256 496 496 1094 **112 328 736]**
	9	[16 16 36 36 94 94 166 166 300 300 **56 124 238 390]**
	10	[8 8 16 16 30 30 46 46 68 68 92 **40 62 86]**
	11	[4 4 6 6 8 8 10 10 12 12 14 14 **16]**
	12	[2 2 2 2 2 2 2 2 2 2 2 2]
14		
	1	[8192 8192]
	2	[4096 4096 8256]
	3	[2048 2048 4096 4096 **6144]**
	4	[1024 1024 2016 2016 11456 **3008]**
	5	[512 512 992 992 6008 6008 **1472 7328]**
	6	[256 256 496 496 3004 3004 6436 **736 3664]**
	7	[128 128 256 256 1420 1420 3004 3004 **384 1744 4588]**
	8	[64 64 136 136 628 628 1288 1288 3474 **784 1948]**
	9	[32 32 72 72 256 256 496 496 1094 1094 **112 328 736 1424]**
	10	[16 16 36 36 94 94 166 166 300 300 456 **56 124 238 390]**
	11	[8 8 16 16 30 30 46 46 68 68 92 92 **24 40 62 86 116]**
	12	[4 4 6 6 8 8 10 10 12 12 14 14 16]
	13	[2 2 2 2 2 2 2 2 2 2 2 2 2]
15		
	1	[16384 16384]
	2	[8192 8192 16512]
	3	[4096 4096 8256 8256 **12416]**
	4	[2048 2048 4096 4096 21000 **6144]**
	5	[1024 1024 2016 2016 11456 11456 **3008 14032]**
	6	[512 512 992 992 6008 6008 12872 **1472 7328]**
	7	[256 256 496 496 3004 3004 6436 6436 **736 3664 9868]**
	8	[128 128 256 256 1420 1420 3004 3004 9950 **384 1744 4588]**
	9	[64 64 136 136 628 628 1288 1288 3474 3474 **784 1948 4464]**
	10	[32 32 72 72 256 256 496 496 1094 1094 1822 **328 736 1424]**
	11	[16 16 36 36 94 94 166 166 300 300 456 456 **56 124 238 390 612]**
	12	[8 8 16 16 30 30 46 46 68 68 92 92 122 **24 40 62 86 116]**
	13	[4 4 6 6 8 8 10 10 12 12 14 14 16 16 **18]**
	14	[2 2 2 2 2 2 2 2 2 2 2 2 2 2]

n	r	$[w_{n-r,n-r-1}, \ldots, w_{n,n-r-1}; \mathbf{w'_{t,n-r-1}}]$
16		
	1	[32768 32768]
	2	[16384 16384 32896]
	3	[8192 8192 16512 16512 **24832]**
	4	[4096 4096 8256 8256 37384 **12416]**
	5	[2048 2048 4096 4096 21000 21000 **6144 25888]**
	6	[1024 1024 2016 2016 11456 11456 24328 **3008 14032]**
	7	[512 512 992 992 6008 6008 12872 12872 **1472 7328 19736]**
	8	[256 256 496 496 3004 3004 6436 6436 26334 **736 3664 9868]**
	9	[128 128 256 256 1420 1420 3004 3004 9950 9950 **384 1744 4588 12524]**
	10	[64 64 136 136 628 628 1288 1288 3474 3474 6206 **208 784 1948 4464]**
	11	[32 32 72 72 256 256 496 496 1094 1094 1822 1822 **328 736 1424 2550]**
	12	[16 16 36 36 94 94 166 166 300 300 456 456 698 **56 124 238 390 612]**
	13	[8 8 16 16 30 30 46 46 68 68 92 92 122 122 **24 40 62 86 116 148]**
	14	[4 4 6 6 8 8 10 10 12 12 14 14 16 16 18 **20]**
	15	[2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]
17		
	1	[65536 65536]
	2	[32768 32768 65536]
	3	[16384 16384 32896 32896 **49408]**
	4	[8192 8192 16512 16512 65536 **24832]**
	5	[4096 4096 8256 8256 37384 37384 **12416 46432]**
	6	[2048 2048 4096 4096 21000 21000 43912 **6144 25888]**
	7	[1024 1024 2016 2016 11456 11456 24328 24328 **3008 14032 37200]**
	8	[512 512 992 992 6008 6008 12872 12872 65536 **1472 7328 19736]**
	9	[256 256 496 496 3004 3004 6436 6436 26334 26334 **736 3664 9868 32340]**
	10	[128 128 256 256 1420 1420 3004 3004 9950 9950 18718 **384 1744 4588 12524]**
	11	[64 64 136 136 628 628 1288 1288 3474 3474 6206 6206 **208 784 1948 4464 8938]**
	12	[32 32 72 72 256 256 496 496 1094 1094 1822 1822 3214 **112 328 736 1424 2550]**
	13	[16 16 36 36 94 94 166 166 300 300 456 456 698 698 **56 124 238 390 612 880]**
	14	[8 8 16 16 30 30 46 46 68 68 92 92 122 122 154 **24 40 62 86 116 148]**
	15	[4 4 6 6 8 8 10 10 12 12 14 14 16 16 18 18 **20]**
	16	[2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]

n	r	$[w_{n-r,n-r-1},\ldots,w_{n,n-r-1};\mathbf{w'_{t,n-r-1}}]$
18		
	1	[131072 131072]
	2	[65536 65536 130816]
	3	[32768 32768 65536 65536 **98304]**
	4	[16384 16384 32896 32896 115312 **16384 49408]**
	5	[8192 8192 16512 16512 65536 65536 **8192 24832 82048]**
	6	[4096 4096 8256 8256 37384 37384 76552 **4096 12416 46432]**
	7	[2048 2048 4096 4096 21000 21000 43912 43912 **6144 25888 66824]**
	8	[1024 1024 2016 2016 11456 11456 24328 24328 155364 **3008 14032 37200]**
	9	[512 512 992 992 6008 6008 12872 12872 65536 65536 **1472 7328 19736 78408]**
	10	[256 256 496 496 3004 3004 6436 6436 26334 26334 51358 **736 3664 9868 32340]**
	11	[128 128 256 256 1420 1420 3004 3004 9950 9950 18718 18718 **384 1744 4588 12524 27486]**
	12	[64 64 136 136 628 628 1288 1288 3474 3474 6206 6206 12598 **208 784 1948 4464 8938]**
	13	[32 32 72 72 256 256 496 496 1094 1094 1822 1822 3214 3214 **112 328 736 1424 2550 4126]**
	14	[16 16 36 36 94 94 166 166 300 300 456 456 698 698 970 **56 124 238 390 612 880]**
	15	[8 8 16 16 30 30 46 46 68 68 92 92 122 122 154 154 **24 40 62 86 116 148 186]**
	16	[4 4 6 6 8 8 10 10 12 12 14 14 16 16 18 18 20]
	17	[2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]
19		
	1	[262144 262144]
	2	[131072 131072 261632]
	3	[65536 65536 130816 130816 **65536 196096]**
	4	[32768 32768 65536 65536 208336 **32768 98304]**
	5	[16384 16384 32896 32896 115312 115312 **16384 49408 145504]**
	6	[8192 8192 16512 16512 65536 65536 130816 **8192 24832 82048]**
	7	[4096 4096 8256 8256 37384 37384 76552 76552 **12416 46432 115720]**
	8	[2048 2048 4096 4096 21000 21000 43912 43912 354332 **6144 25888 66824]**
	9	[1024 1024 2016 2016 11456 11456 24328 24328 155364 155364 **3008 14032 37200 181136]**
	10	[512 512 992 992 6008 6008 12872 12872 65536 65536 130816 **1472 7328 19736 78408]**
	11	[256 256 496 496 3004 3004 6436 6436 26334 26334 51358 51358 **736 3664 9868 32340 76382]**
	12	[128 128 256 256 1420 1420 3004 3004 9950 9950 18718 18718 43606 **384 1744 4588 12524 27486]**
	13	[64 64 136 136 628 628 1288 1288 3474 3474 6206 6206 12598 12598 **208 784 1948 4464 8938 16270]**
	14	[32 32 72 72 256 256 496 496 1094 1094 1822 1822 3214 3214 4846 **112 328 736 1424 2550 4126]**
	15	[16 16 36 36 94 94 166 166 300 300 456 456 698 698 970 970 **56 124 238 390 612 880 1242]**
	16	[8 8 16 16 30 30 46 46 68 68 92 92 122 122 154 154 192 **24 40 62 86 116 148 186]**
	17	[4 4 6 6 8 8 10 10 12 12 14 14 16 16 18 18 20 20 **22]**
	18	[2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]

n	r	$[w_{n-r,n-r-1},\ldots,w_{n,n-r-1};\mathbf{w'_{t,n-r-1}}]$
20		
	1	[524288 524288]
	2	[262144 262144 523776]
	3	[131072 131072 261632 261632 **392192]**
	4	[65536 65536 130816 130816 394384 **196096]**
	5	[32768 32768 65536 65536 208336 208336 **98304 264640]**
	6	[16384 16384 32896 32896 115312 115312 223840 **49408 145504]**
	7	[8192 8192 16512 16512 65536 65536 130816 130816**24832 82048 196096]**
	8	[4096 4096 8256 8256 37384 37384 76552 76552 783276 **12416 46432 115720]**
	9	[2048 2048 4096 4096 21000 21000 43912 43912 354332 354332 **6144 25888 66824 403224]**
	10	[1024 1024 2016 2016 11456 11456 24328 24328 155364 155364 314280 **3008 14032 37200 181136]**
	11	[512 512 992 992 6008 6008 12872 12872 65536 65536 130816 130816 **1472 7328 19736 78408 196096]**
	12	[256 256 496 496 3004 3004 6436 6436 26334 26334 51358 51358 136630 **736 3664 9868 32340 76382]**
	13	[128 128 256 256 1420 1420 3004 3004 9950 9950 18718 18718 43606 43606 **384 1744 4588 12524 27486 56254]**
	14	[64 64 136 136 628 628 1288 1288 3474 3474 6206 6206 12598 12598 20350 **208 784 1948 4464 8938 16270]**
	15	[32 32 72 72 256 256 496 496 1094 1094 1822 1822 3214 3214 4846 4846 **112 328 736 1424 2550 4126 6478]**
	16	[16 16 36 36 94 94 166 166 300 300 456 456 698 698 970 970 1352 **56 124 238 390 612 880 1242]**
	17	[8 8 16 16 30 30 46 46 68 68 92 92 122 122 154 154 192 192 **24 40 62 86 116 148 186 226]**
	18	[4 4 6 6 8 8 10 10 12 12 14 14 16 16 18 18 20 20 22]
	19	[2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]
n = 21		
	1	[1048576 1048576]
	2	[524288 524288 1048576]
	3	[262144 262144 523776 523776 **785408]**
	4	[131072 131072 261632 261632 788768 **392192]**
	5	[65536 65536 130816 130816 394384 394384 **196096 502912]**
	6	[32768 32768 65536 65536 208336 208336 394384 **98304 264640]**
	7	[16384 16384 32896 32896 115312 115312 223840 223840 **49408 145504 332368]**
	8	[8192 8192 16512 16512 65536 65536 130816 130816 1687676 **24832 82048 196096]**
	9	[4096 4096 8256 8256 37384 37384 76552 76552 783276 783276 **12416 46432 115720 872424]**
	10	[2048 2048 4096 4096 21000 21000 43912 43912 354332 354332 721260 **6144 25888 66824 403224]**
	11	[1024 1024 2016 2016 11456 11456 24328 24328 155364 155364 314280 314280 **3008 14032 37200 181136 473196]**
	12	[512 512 992 992 6008 6008 12872 12872 65536 65536 130816 130816 394384 **1472 7328 19736 78408 196096]**
	13	[256 256 496 496 3004 3004 6436 6436 26334 26334 51358 51358 136630 136630 **736 3664 9868 32340 76382 175390]**
	14	[128 128 256 256 1420 1420 3004 3004 9950 9950 18718 18718 43606 43606 74614 **384 1744 4588 12524 27486 56254]**
	15	[64 64 136 136 628 628 1288 1288 3474 3474 6206 6206 12598 12598 20350 20350 **208 784 1948 4464 8938 16270 28102]**
	16	[32 32 72 72 256 256 496 496 1094 1094 1822 1822 3214 3214 4846 4846 7548 **112 328 736 1424 2550 4126 6478]**
	17	[16 16 36 36 94 94 166 166 300 300 456 456 698 698 970 970 1352 1352 **56 124 238 390 612 880 1242 1658]**
	18	[8 8 16 16 30 30 46 46 68 68 92 92 122 122 154 154 192 192 232 **24 40 62 86 116 148 186 226]**
	19	[4 4 6 6 8 8 10 10 12 12 14 14 16 16 18 18 20 20 22 22 **24]**
	20	[2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]

3.3.1. The weights in some cases where $a_1,\ldots,a_t$ are linearly dependent

We have seen that restricting ourselves to the case where $a_1,\ldots,a_t$ are linearly independent over $\mathbb{F}_2$ reduces the number of the weights, which can be found by using Theorem 1, because t is then necessarily in the range $\{n-r,\ldots,n\}$ and since, for fixed n and r (i.e., for fixed n and s), all the obtained functions corresponding to the same t have the same Hamming weight. In the present section, we investigate two cases where $a_1,\ldots,a_t$ are linearly dependent. We shall see that the first does not provide more weights but the second does.

Case where two elements are equal: We study this case by curiosity, to check whether with t elements $a_1,\ldots,a_t$, it is identical to the case of $t-2$ elements or not (the formulas are different but the functions and/or the weights may be the same).

To ease the comparison, we start with $t+2$ elements $a_1,\ldots,a_{t+2}$ such that (without loss of generality) $a_{t+2} = a_{t+1}$, then for every $J \subseteq \{1,\ldots,t+2\}$, we have that $\sum_{i\in J} a_i$ equals $\begin{cases} \sum_{i\in J} a_i \text{ if } \{t+1,t+2\}\cap J=\emptyset \\ \sum_{i\in J\setminus\{t+1,t+2\}} a_i \text{ if } \{t+1,t+2\}\subseteq J \end{cases}$. We get the same atomic function (which cancels then) if exactly one element among $\{t+1,t+2\}$ belongs to J, whether we choose $t+1$ or $t+2$.
We deduce that:

$$f^{(s)}_{a_1,\ldots,a_{t+2}} = \delta_{\sum_{i=1}^{t} a_i} +$$

$$\sum_{j=0}^{s} \mu_{t+2,s}(j) \sum_{J\subseteq\{1,\ldots,t\};|J|=j} \delta_{\sum_{i\in J} a_i} + \sum_{j=0}^{s-2} \mu_{t+2,s}(j+2) \sum_{J\subseteq\{1,\ldots,t\};|J|=j} \delta_{\sum_{i\in J} a_i},$$

that is: $f^{(s)}_{a_1,\ldots,a_{t+2}} =$

$$\delta_{\sum_{i=1}^{t} a_i} + \sum_{j=0}^{s} \Big(\mu_{t+2,s}(j) + \mu_{t+2,s}(j+2)\Big) \sum_{J\subseteq\{1,\ldots,t\};|J|=j} \delta_{\sum_{i\in J} a_i}, \tag{3.4}$$

where $\mu_{t,s}(j)$ equals $\binom{t-j-1}{s-j} \bmod 2$ if $0 \le j \le s$ and equals 0 otherwise.
For $s < t$, we obtain $f^{(s)}_{a_1,\ldots,a_{t+2}} = f^{(s)}_{a_1,\ldots,a_t}$. Indeed, $\mu_{t+2,s}(j) + \mu_{t+2,s}(j+2) = \left(\binom{t-j+1}{s-j} + \binom{t-j-1}{s-j-2}\right) \bmod 2$ equals $\binom{t-j-1}{s-j} \bmod 2 = \mu_{t,s}(j)$.

Remark 3. *We could additionally consider the cases $s = t$ and $s = t+1$ since, having $s < t+2$, we can still use Relation (3.2), with $t+2$ in the place of t. The only difference with the case $s < t$ is that the atomic function $\delta_{\sum_{i=1}^{t} a_i}$ may equal one of the other atomic functions appearing in (3.4). When we evaluate the Hamming weight, we then have to consider particular cases according to whether $\delta_{\sum_{i=1}^{t} a_i}$ is present in $\sum_{j=0}^{s} \Big(\mu_{t+2,s}(j) + \mu_{t+2,s}(j+2)\Big) \sum_{J\subseteq\{1,\ldots,t\};|J|=j} \delta_{\sum_{i\in J} a_i}$ (that is, in $\sum_{j=s-1}^{s} \Big(\mu_{t+2,s}(j) + \mu_{t+2,s}(j+2)\Big) \sum_{J\subseteq\{1,\ldots,t\};|J|=j} \delta_{\sum_{i\in J} a_i}$), and then cancels, or not. Anyway, the sum $\sum_{j=0}^{s} \Big(\mu_{t+2,s}(j) + \mu_{t+2,s}(j+2)\Big)\binom{s}{j}$ is smaller than or equal to $\sum_{j=0}^{s} \binom{s}{j} = 2^s$ and the sum $\sum_{j=0}^{s} \Big(\mu_{t+2,s}(j) + \mu_{t+2,s}(j+2)\Big)\binom{s-1}{j}$ is still smaller, and we will necessarily obtain 0 since 2^s is strictly smaller than the minimum distance of $RM(r,n)$.*

Case where one element equals the sum of two others: We start with $t+1$ elements $a_1, \ldots, a_{t+1}$ such that (without loss of generality) $a_{t+1} = a_t + a_{t-1}$ (and $t \ge 3$, so that there remains one element after cancellation in the sum $\sum_{i=1}^{y} a_i$). For every $J \subseteq \{1, \ldots, t+1\}$, we have that $\sum_{i\in J} a_i$ equals:

$$\begin{cases} \sum_{i\in J} a_i \text{ if } t+1 \notin J \\ \sum_{i\in J\cup\{t-1,t\}\setminus\{t+1\}} a_i \text{ if } \{t-1,t,t+1\} \cap J = \{t+1\} \\ \sum_{i\in J\cup\{t\}\setminus\{t-1,t+1\}} a_i \text{ if } \{t-1,t,t+1\} \cap J = \{t-1,t+1\} \\ \sum_{i\in J\cup\{t-1\}\setminus\{t,t+1\}} a_i \text{ if } \{t-1,t,t+1\} \cap J = \{t,t+1\} \\ \sum_{i\in J\setminus\{t-1,t,t+1\}} a_i \text{ if } \{t-1,t,t+1\} \subseteq J \end{cases},$$

then:

$$f^{(s)}_{a_1,\ldots,a_{t+1}} = \delta_{\sum_{i=1}^{t-2} a_i} + \sum_{j=0}^{s} \mu_{t+1,s}(j) \sum_{J\subseteq\{1,\ldots,t\};|J|=j} \delta_{\sum_{i\in J} a_i} +$$

$$\sum_{j=1}^{s} \mu_{t+1,s}(j) \sum_{J\subseteq\{1,\ldots,t-2\};|J|=j-1} \delta_{a_t+a_{t-1}+\sum_{i\in J} a_i} +$$

$$\sum_{j=0}^{s} \mu_{t+1,s}(j) \sum_{J\subseteq\{1,\ldots,t-2\};|J|=j-2} \delta_{a_t+\sum_{i\in J} a_i} +$$

$$\sum_{j=1}^{s} \mu_{t+1,s}(j) \sum_{J\subseteq\{1,\ldots,t-2\};|J|=j-2} \delta_{a_{t-1}+\sum_{i\in J} a_i} +$$

$$\sum_{j=1}^{s}\mu_{t+1,s}(j)\sum_{J\subseteq\{1,\ldots,t-2\};|J|=j-3}\delta_{\sum_{i\in J}a_i}=$$

$$\delta_{\sum_{i=1}^{t-2}a_i}+\sum_{j=0}^{s}\Big(\mu_{t+1,s}(j)+\mu_{t+1,s}(j+3)\Big)\sum_{J\subseteq\{1,\ldots,t-2\};|J|=j}\delta_{\sum_{i\in J}a_i}+$$

$$\sum_{j=0}^{s}\Big(\mu_{t+1,s}(j)+\mu_{t+1,s}(j+1)\Big)\sum_{\substack{J\subseteq\{1,\ldots,t\};|J|=j\\ J\cap\{t-1,t\}=\{t-1\}}}\delta_{\sum_{i\in J}a_i}+$$

$$\sum_{j=0}^{s}\Big(\mu_{t+1,s}(j)+\mu_{t+1,s}(j+1)\Big)\sum_{\substack{J\subseteq\{1,\ldots,t\};|J|=j\\ J\cap\{t-1,t\}=\{t\}}}\delta_{\sum_{i\in J}a_i}+$$

$$\sum_{j=0}^{s+1}\Big(\mu_{t+1,s}(j)+\mu_{t+1,s}(j-1)\Big)\sum_{\substack{J\subseteq\{1,\ldots,t\};|J|=j\\ \{t-1,t\}\subseteq J}}\delta_{\sum_{i\in J}a_i},$$

where $\mu_{t,s}(j)$ equals $\binom{t-j-1}{s-j} \bmod 2$ if $0\le j\le s$ and equals 0 otherwise.

Proposition 4. *Let $n\ge 3$, $s\ge 0$, and $t\ge 3$ be integers such that $s<t$ and $s<n$. For any elements $a_1,\ldots,a_t$ of $\mathbb{F}_2^n$, let $f^{(s)}_{a_1,\ldots,a_{t+1}}$ be the Boolean function given by (3.2) with $a_{t+1}=a_t+a_{t-1}$. If $a_1,\ldots,a_t$ are linearly independent over $\mathbb{F}_2$, then $f^{(s)}_{a_1,\ldots,a_{t+1}}$ has Hamming weight:*

$$w'_{t,s}=\epsilon+\sum_{j=0}^{s}\Big(\mu_{t+1,s}(j)+\mu_{t+1,s}(j+3)\Big)\binom{t-2}{j}+$$

$$2\sum_{j=1}^{s}\Big(\mu_{t+1,s}(j)+\mu_{t+1,s}(j+1)\Big)\binom{t-2}{j-1}+$$

$$\sum_{j=2}^{s+1}\Big(\mu_{t+1,s}(j)+\mu_{t+1,s}(j-1)\Big)\binom{t-2}{j-2},$$

where $\mu_{t+1,s}(j)$ equals $\binom{t-j}{s-j} \bmod 2$ if $0\le j\le s$ and equals 0 otherwise (and the additions "$\mu_{t+1,s}(j)+\mu_{t+1,s}(j+3)$", etc., are made modulo 2). Here, ϵ equals -1 if $s\ge t-2$ and $\mu_{t+1,s}(t-2)=1$, and equals 1 otherwise. If $a_1,\ldots,a_t$ are linearly dependent, then $f^{(s)}_{a_1,\ldots,a_{t+1}}$ has a Hamming weight of at most $w'_{t,s}$.

The value of ϵ is -1 when $\delta_{\sum_{i=1}^{t-2}a_i}$ is equal to one of the other atomic functions present in the formula, and this is possible only when $s\ge t-2$ and $\mu_{t+1,s}(t-2)+\mu_{t+1,s}(t+1)=1$, that is, $\mu_{t+1,s}(t-2)=1$.

We obtain again the weights that we had obtained with linearly independent vectors $a_1,\ldots,a_t$, and we obtain about 50% additional weights, which we display in bold (after these ones) in Table 1.

3.3.2. Making linear combinations of the constructed functions

We have seen in the previous subsection that taking $a_1,\ldots,a_t$ linearly dependent provides more weights than taking them linearly independent, but not in a large proportion (at least for the two cases that we studied: two elements equal and one element equal to the sum of two others). We need then

to find other ways to provide more weights. One is very simple. Since all the functions $f^{(s)}_{a_1,\dots,a_t}$ have an algebraic degree of at most $r = n - s - 1$, we can sum, for every choice of s, some of the functions $f^{(s)}_{a_1,\dots,a_t}, f^{(s+1)}_{a_1,\dots,a_t}, \dots, f^{(t-1)}_{a_1,\dots,a_t}$ for different choices of $t > s$ and of $a_1, \dots, a_t$.

The difficulty is to evaluate the Hamming weight of the resulting functions, but there is a case where the weight is easily determined: when we take disjoint families of vectors a_i whose union is made of linearly independent vectors.

In the simplest case, we have (globally) t linearly independent vectors $a_1, \dots, a_t$ in $\mathbb{F}_2^n$ (with $t \le n$), and we partition $\{1, \dots, t\}$ in two subsets (without loss of generality, we can take these subsets equal to $\{1, \dots, l\}$ and $\{l+1, \dots, t\}$), then two functions $f^{(s)}_{a_1,\dots,a_l}$ and $f^{(s')}_{a_{l+1},\dots,a_t}$ with $s < l$ and $s' < t - l$ have algebraic degrees of at most $r = n - s - 1$ and $r' = n - s' - 1$, respectively, and they have disjoint supports. Their sum has then an algebraic degree of at most $\max(r, r')$ and has for Hamming weight the sum of their Hamming weights, that is, $w_{l,s} + w_{t-l,s'}$.

Example 3. *Let us take $n = t = 12$ and $s = s' = 2$, that is, $r = r' = 9$. We must take l and $12 - l$ strictly larger than 2, that is, l between 3 and 9. We find in Table 1 that, when l ranges from 3 to 9, $(w_{l,2}, w_{t-l,2})$ takes the following values, indicated in the row corresponding to $n = 12$ and $r = 9$: $(8, 46), (8, 30), (16, 30), (16, 16), (30, 16), (30, 8), (46, 8)$, respectively. We obtain then the weights $54, 38, 46, 32, 46, 38, 54$ in $RM(9, 12)$. This way, we obtain two new weights (38 and 54) in $RM(9, 12)$.*

Example 3 can be generalized:

Proposition 5. *Let n and $r < n$ be any positive integers. All the numbers $w_{l,n-r_1-1} + w_{t-l,n-r_2-1}$, where $t \le n$, $n - r_1 \le l \le t - n + r_2$, $r_1 \le r$, $r_2 \le r$ and $r_1 + r_2 \ge 2n - t$ are weights in $RM(r, n)$.*

This is straightforward. The condition $l \le t - n + r_2$ is for having $t - l \ge n - r_2$ and the condition $r_1 + r_2 \ge 2n - t$ is for having $n - r_1 \le t - n + r_2$. The condition $l \le t$ is automatically satisfied thanks to $l \le t - n + r_2$.

Of course, Proposition 5 can be generalized to sums of more than two numbers (taking more than two families partitioning $\{a_1, \dots, a_t\}$).

Remark 4. *The conditions $r \ge r_1, r_2$ and $r_1 + r_2 \ge 2n - t$ imply that $2r \ge 2n - t$, that is, $t \ge 2n - 2r$ and since t cannot be larger than n, this means that Proposition 5 can be used only if $n \ge 2n - 2r$, that is, $r \ge \frac{n}{2}$. Our results are then unfortunately limited to those of Proposition 3 (that is, those of Table 1) for the orders in the lower half of $[0, n]$ (those for which the table provides the least values), in particular for the smallest order for which the weight spectrum is unknown: $r = 3$. We shall see below that, on the contrary, we can derive a very large number of weights as soon as r is large enough.*
Note that if we partition $\{a_1, \dots, a_t\}$ in three families $\{a_1, \dots, a_l\}$, $\{a_{l+1}, \dots, a_k\}$ and $\{a_{k+1}, \dots, a_t\}$, we get the weight $w_{l,n-r_1-1} + w_{k-l,n-r_2-1} + w_{t-k-l,n-r_3-1}$ (instead of $w_{l,n-r_1-1} + w_{t-l,n-r_2-1}$ that we had with two families), and conditions $l \ge n - r_1$, $k - l \ge n - r_2$ and $t - k - l \ge n - r_3$ imply $t \ge 3n - (r_1 + r_2 + r_3)$, that is, $3r \ge r_1 + r_2 + r_3 \ge 3n - t \ge 2n$, and the condition $r \ge \frac{n}{2}$ becomes $r \ge \frac{2n}{3}$.

For each $n \le 23$, the weights provided by Proposition 5 can be obtained by adding in Table 1 any number located in any row $r_1 \le r$ at the k-th position (by taking $k = l - (n - r_1) + 1$ so that it starts at position 1 in the list given by Table 1) where $k \le t - n + r_2 - (n - r_1) + 1 = t - 2n + r_1 + r_2 + 1$, and the number located in any row $r_2 \le r$ such that $r_1 + r_2 \ge 2n - t$, at the position corresponding to $t - l$, that is at the position $t - (k + n - r_1 - 1) - (n - r_2) + 1 = t - k - 2n + r_1 + r_2 + 2$. It is for large orders that our

method gives the best results, since the weights are then more numerous in Table 1; and making sums (applying Proposition 5) is also possible only when $r \geq \frac{n}{2}$, and these sums are more numerous when r is larger.

Example 3, revisited: For $n = 12$ and $r = 9$, the condition $t \geq 2n - 2r$ writes $t \geq 6$, and we obtain the following:

For $t = 12$: The conditions $r_1, r_2 \leq r$ and $r_1 + r_2 \geq 2n - t$ allow $(r_1, r_2) = (3, 9); (4, 8), (4, 9); (5, 7), (5, 8), (5, 9); \ldots; (9, 3), (9, 4), \ldots, (9, 9)$, which provide the following weights:
For (3,9): 520
for (4,8): 272
for (4,9): 264 and 264
for (5,7): 160
for (5, 8): 144 and 144
for (5, 9): 144 and 136 and 264
for (6,6): 128
for (6,7): 96 and 96
for (6,8): 100 and 80 and 152
for (6,9): 80 and 80 and 144 and 144
for(7,5), we have the same as for the swap (5,7)
for (7,6), we have the same as for the swap
for (7,7): 104 and 64 and 104
for (7,8): 68 and 68 and 88 and 88
for (7,9): 62 and 48 and 88 and 80 and 264
for (8,4), (8,5), (8,6), (8,7), we have the same as for their swaps,
for (8,8): 110 and 52 and 72 and 52 and 110
for (8,9): 46 and 46 and 52 and 52 and 102 and 102
for (9,3), (9,4), (9,5), (9,6), (9,7), (9,8) we obtain the same as for their swaps
for (9,9): 54 and 38 and 46 and 32 and 46 and 38 and 54.
For this single value of t, we have then obtained in addition to the weights present in Table 1 and to the two weights 38 and 54 found above: 48, 52, 62, 72, 80, 88, 96, 100, 102, 104, 110, 136, 144, 152, 160, 264, 272, 520.

We would also have to consider all the other values of t, from 11 down to 6.

As mentioned after Proposition 5, we could find more weights by partioning $\{a_1, \ldots, a_t\}$ in more than two families.

We see that our method gives in fact a large number of weights, for sufficiently large orders, that is, when the usual constructions of Boolean functions (Maiorana-McFarland, etc.) give the worst results. It is then nicely complementary to the method using classic constructions.

Example 4. *We have seen in introduction that trying to determine the weight spectrum of all codes $RM(n - c, n)$ for $c = 6$ (the smallest value of c for which this is an open problem) requires determining the weights in $RM(6 + k, 12 + k)$, for some $k \geq 0$.*

– Let us first consider $RM(6, 12)$. Table 1 already provides the weights 64, 128, 136, 208, 256, 384, 496, 512, 628, 736, 784, 992, 1024, 1288, 1420, 1472, 1744, 2016, 2048, 3004. We could complete with the weights provided by Kasami-Tokura. We could also add the weights obtained by adding 16 weights from $RM(2, 8)$ (that are 0, 64, 96,112,120,128,136, 144, 160, 192, 256

according to what we know on quadratic functions) and adding 8 weights of $RM(3,9)$ (that are 0, 64, 96,112,120,128,136,144,148,152,156,160,164,168,172,176,180,184,188, 192,196,200,204,208,212,216,220,224,228,232,236,240,244,248,252,256,260,264, 268,272,276,280,284,288,292,296,300,304,308,312,316,320, 324,328,332,336,340, 344,348,352,356,360,364,368,376,384,392,400,416,448,512, see the URL: https://isec.ec.okayama-u.ac.jp/home/kusaka/wd/RM/tomita/RM512_130.wd).
The weights of $RM(4,10)$ and $RM(5,11)$ are unknown (but we know from Table 1 that they include respectively 64,128,136,208,256,384,496,512,628 and 64,128,136,208,256,384,496,512,628,736,784,992,1024,1420. We can add the weights obtained by adding weights from $RM(r_1,12)$ and $RM(r_2,12)$ in Table 1 as explained above. For $n = 12$ and $r = 6$, the condition $n \geq t \geq 2n - 2r$ writes $t = 12$. The conditions $r_1, r_2 \leq r$ and $r_1 + r_2 \geq 2n - t$ allow only one possibility: $(r_1, r_2) = (6,6)$, which provides only one weight (since the number $k \in \{1, t - 2n + r_1 + r_2 + 1\}$ in the description we gave can take value 1 only): 128, which is already there.
– Let us then consider $RM(7,13)$. Table 1 already provides the weights 64,128, 136, 208, 256, 384, 496, 512, 628, 736, 784, 992,1024, 1288,1420, 1472, 1744, 1948, 2016, 2048, 3004, 3008, 3664, 4096, 6008. Note also that the weights of $RM(6,12)$ are also weights of $RM(7,13)$, but this provides no new weight. We could complete with the weights provided by Kasami-Tokura, and we can add the weights obtained by adding weights from $RM(r_1,13)$ and $RM(r_2,13)$ in Table 1 as explained above. For $n = 13$ and $r = 7$, the condition $n \geq t \geq 2n - 2r$ writes $t \in \{12, 13\}$.
- For $t = 13$, the conditions $r_1, r_2 \leq r$ and $r_1 + r_2 \geq 2n - t$ allow (up to a swap between r_1 and r_2): $(r_1, r_2) = (7,6), (7,7)$. The case (7,6) provides one weight (the number $k \in \{1, t - 2n + r_1 + r_2 + 1\}$ can take value 1 only): 96. The case (7,7) provides one weight (the number $k \in \{1, t - 2n + r_1 + r_2 + 1\}$ can take values 1 and 2, which give the same weight): 64 which was already there.
- For $t = 12$, the conditions $r_1, r_2 \leq r$ and $r_1 + r_2 \geq 2n - t$ allow: $(r_1, r_2) = (7,7)$, which provides the weight 64 also already obtained.
We could continue by visiting $RM(8,14)$ (which is the first case where we obtain a weight not divisible by 4: 3474) etc., but with this example, we see the huge difference between low and high orders.
We leave as an open problem the determination of more weights in $RM(6,12)$ (and in particular, some that are not divisible by 4), which will probably need to find another method than exploiting Relation (3.1).

3.4. The ANF of the constructed functions when $a_1, \ldots, a_t$ are linearly independent over $\mathbb{F}_2$

We have seen that for $t \leq n$, two choices "$a_1, \ldots, a_t$", respectively "$a'_1, \ldots, a'_t$", of linearly independent elements give linearly equivalent functions, having then the same weight and the same algebraic degree.

Let us then determine the ANF of $f^{(s)}_{e_1,\ldots,e_t}$, where $t \leq n$. We shall need the following lemma:

Lemma 2. *Let $n \geq 1, t \geq 1, j \geq 0$ be integers such that $j \leq t \leq n$ and let $e_1, \ldots, e_t$ be the t first elements of the canonical basis of $\mathbb{F}_2^n$. The Boolean function:*

$$h_{j,e_1,\ldots,e_t} = \sum_{J \subseteq \{1,\ldots,t\}; |J| = j} \delta_{\sum_{i \in J} e_i}$$

has for ANF:

$$\sum_{\substack{I\subseteq\{1,\dots,n\}\\ j\preceq|\{1,\dots,t\}\cap I|}} x^I,$$

where $|\dots|$ *denotes the size and* $j \preceq m$ *means that if* $j = \sum_{k\in K} 2^k$ *and* $m = \sum_{l\in L} 2^l$ *are the binary expansions of* j *and* m*, then* $K \subseteq L$*.*

Proof. Since $\sum_{i\in J} e_i$ is the vector of support J, we have, denoting $x = (x_1, \dots, x_n)$:

$$\begin{aligned} h_{j,e_1,\dots,e_t}(x) &= \sum_{J\subseteq\{1,\dots,t\};|J|=j} \Big(\prod_{k\in J} x_k\Big)\Big(\prod_{k\in\{1,\dots,n\}\setminus J} (x_k+1)\Big)\\ &= \sum_{J\subseteq\{1,\dots,t\};|J|=j}\ \sum_{J\subseteq I\subseteq\{1,\dots,n\}} x^I\\ &= \sum_{I\subseteq\{1,\dots,n\}}\ \sum_{J\subseteq\{1,\dots,t\}\cap I;|J|=j} x^I\\ &= \sum_{I\subseteq\{1,\dots,n\}} \binom{|\{1,\dots,t\}\cap I|}{j} x^I, \end{aligned}$$

where these sums are taken modulo 2 and $\binom{|\{1,\dots,t\}\cap I|}{j}$ equals 0 if $|\{1,\dots,t\}\cap I| < j$. The proof is complete thanks to Lucas' theorem [22, page 404]. □

We deduce:

Proposition 6. *Let* $n \geq 1$*,* $s \geq 0$ *and* $t \geq 1$ *be integers such that* $s < t$ *and* $s < n$*. Given any linearly independent elements* $a_1, \dots, a_t$ *of* $\mathbb{F}_2^n$*, the Boolean function (3.2) is linearly equivalent to the function of ANF:*

$$\sum_{\{1,\dots,t\}\subseteq I} x^I + \sum_{j=0}^{s} \mu_{t,s}(j) \sum_{\substack{I\subseteq\{1,\dots,n\}\\ j\preceq|\{1,\dots,t\}\cap I|}} x^I,$$

where $\mu_{t,s}(j) = \binom{t-j-1}{s-j} \bmod 2 = \binom{t-j-1}{t-s-1} \bmod 2$*.*

This is straightforward since $f^{(s)}_{e_1,\dots,e_t} = h_{t,e_1,\dots,e_t} + \sum_{j=0}^{s} \mu_{t,s}(j)\, h_{j,e_1,\dots,e_t}$. □

Remark 5. *Even after these caculations, it is not obvious to see (what we already know) that* $f^{(s)}_{e_1,\dots,e_t}$ *has an algebraic degree of at most* $r = n - s - 1$*, that is, for every* $I \subseteq \{1,\dots,n\}$ *whose size is strictly larger than* r*, we have* $\sum_{\substack{0\leq j\leq s\\ j\preceq|\{1,\dots,t\}\cap I|}} \mu_{t,s}(j) = 1 \bmod 2$ *if* $\{1,\dots,t\} \subseteq I$ *and* $\sum_{\substack{0\leq j\leq s\\ j\preceq|\{1,\dots,t\}\cap I|}} \mu_{t,s}(j) = 0 \bmod 2$ *otherwise.*

Open problem: Determine the exact algebraic degree of $f^{(s)}_{a_1,\dots,a_t}$ by means of s, n and $a_1, \dots, a_t$.
Subproblem: Determine the algebraic degree of $f^{(s)}_{a_1,\dots,a_t}$ by means of s and n when $a_1, \dots, a_t$ are linearly independent.

Still more complex is the following:
Open problem: Determine what can be the ANF of $f^{(s)}_{a_1,\dots,a_t}$ when $a_1, \dots, a_t$ are linearly dependent.

4. Conclusions

We have introduced a novel way of constructing Reed-Muller codewords. It consists of exploiting relations satisfied by all n-variable Boolean or vectorial functions F of an algebraic degree of at most s (corresponding when F is Boolean to codewords in $RM(s, n)$), these relations being interpretable in terms of the orthogonality between some Boolean function, say f, and (the coordinate functions of) all such F. Function f belongs then to $RM(r, n)$, where $r = n - s - 1$. This construction depends on n, s (or r), a parameter t and the choice of t vectors a_i. We showed how it allows us to determine weights in Reed-Muller codes that are not accessible by other methods, as far as we know, and in a simpler way. As a matter of fact, our method for determining weights in Reed-Muller codes is complementary of the classic method, which consists of using the known constructions, since the latter is more efficient for low orders and our method is more efficient for large orders. Anyway, the method using the known constructions poses technical problems (and provides a number of weights that is small compared to the amount of work needed) while ours provides weights with less difficulties. Functions having the weights we can derive with our method can be deduced, as well as a general form of their ANF when the vectors a_i are linearly independent, but determining mathematically their exact algebraic degree seems difficult. This is one of the open problems we proposed. We also found more weights by considering cases where the vectors are linearly dependent. We could also identify that with some of the constructed functions having disjoint supports, the weights of the sums are equal to the sums of the weights; this provided for each Reed-Muller code of a sufficiently large order a very large number of new weights.

More work is possible in many directions, for instance, by investigating as many cases of functions as possible where the vectors a_i are linearly dependent and studying sums of such functions as well. Moreover, there may be other relations to find that are interpretable in terms of orthogonality, leading to more codewords and weights in Reed-Muller codes. This may provide an avenue for further results, with the ultimate goal of determining all the weight spectra of Reed-Muller codes (starting with those of high orders when they are still unknown, since they seem to be more accessible than those of low orders larger than 2), and still better, their weight distributions.

Use of AI tools declaration

The author declares he has not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

We deedply thank Stjepan Picek for his kind help with the tables; we are indebted to him.
The research of the author is partly supported by the Norwegian Research Council.

Conflict of interest

The author declares no conflict of interest.

References

1. E. Abbe, O. Sberlo, A. Shpilka, M. Ye, Reed-Muller codes, *Found. Trends Commun.*, **20** (2023), 1–156. https://doi.org/10.1561/0100000123
2. E. Abbe, A. Shpilka, A. Wigderson, Reed-Muller codes for random erasures and errors, *IEEE T. Inform. Theory*, **61** (2015), 5229–5252. https://doi.org/10.1109/TIT.2015.2462817
3. C. Beierle, A. Biryukov, A. Udovenko, On degree-*d* zero-sum sets of full rank, *Cryptogr. Commun.*, **12** (2020), 685–710. https://doi.org/10.1007/s12095-019-00415-0
4. E. R. Berlekamp, N. J. A. Sloane, Restrictions on the weight distributions of the Reed-Muller codes, *Inform. Control*, **14** (1969), 442–446. https://doi.org/10.1016/S0019-9958(69)90150-8
5. E. R. Berlekamp, N. J. A. Sloane, Weight enumerator for second-order Reed-Muller codes, *IEEE T. Inform. Theory*, **16** (1970), 745–751. https://doi.org/10.1109/TIT.1970.1054553
6. Y. L. Borissov, *On McEliece's result about divisibility of the weights in the binary Reed-Muller codes*, In: Seventh International Workshop on Optimal Codes and Related Topics, 2013, 47–52. Available from: `http://www.moi.math.bas.bg/oc2013/a7.pdf`.
7. C. Carlet, *A transformation on Boolean functions, its consequences on some problems related to Reed-Muller codes*, In: Proceedings of EUROCODE 1990, Lecture Notes in Computer Science, **514** (1991), 42–50. https://doi.org/10.1007/3-540-54303-1_116
8. C. Carlet, *Boolean functions for cryptography and coding theory*, Cambridge University Press, 2021.
9. C. Carlet, The weight spectrum of the Reed-Muller codes $RM(m-5,m)$, *IEEE T. Inform. Theory*, 2024. http://dx.doi.org/10.1109/TIT.2023.3343697
10. C. Carlet, P. Méaux, A complete study of two classes of Boolean functions: Direct sums of monomials and threshold functions, *IEEE T. Inform. Theory*, **68** (2022), 3404–3425. https://doi.org/10.1109/TIT.2021.3139804
11. C. Carlet, E. Prouff, M. Rivain, T. Roche, *Algebraic decomposition for probing security*, In: Proceedings of CRYPTO 2015, Lecture Notes in Computer Science, **9215** (2015), 742–763. https://doi.org/10.1007/978-3-662-47989-6_36
12. C. Carlet, P. Solé, The weight spectrum of two families of Reed-Muller codes, *Discrete Math.*, **346** (2023), 113568. https://doi.org/10.1016/j.disc.2023.113568
13. C. Ding, C. Li, Y. Xia, Another generalisation of the binary Reed-Muller codes and its applications, *Finite Fields Th. Appl.*, **53** (2018), 144–174. https://doi.org/10.1016/j.ffa.2018.06.006
14. Final report of 3GPP TSG RAN WG1 #87 v1.0.0. Available from: `http://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_87/Report/`.
15. T. Kasami, The weight enumerators for several classes of subcodes of the 2nd order binary Reed-Muller codes, *Inform. Control*, **18** (1971), 369–394. https://doi.org/10.1016/S0019-9958(71)90473-6
16. T. Kasami, N. Tokura, On the weight structure of the Reed Muller codes, *IEEE T. Inform. Theory*, **16** (1970), 752–759. https://doi.org/10.1109/TIT.1970.1054545

17. T. Kasami, N. Tokura, S. Azumi, On the weight enumeration of weights less than 2.5*d* of Reed-Muller codes, *Inform. Control*, **30** (1976), 380–395. https://doi.org/10.1016/S0019-9958(76)90355-7

18. T. Kaufman, S. Lovett, E. Porat, Weight distribution and list-decoding size of Reed-Muller codes, *IEEE T. Inform. Theory*, **58** (2012), 2689–2696. https://doi.org/10.1109/TIT.2012.2184841

19. A. M. Kerdock, A class of low-rate non linear codes, *Inform. Control*, **20** (1972), 182–187. https://doi.org/10.1016/S0019-9958(72)90376-2

20. S. J. Lin, Y. S. Han, N. Yu, New locally correctable codes based on projective Reed-Muller codes, *IEEE T. Inform. Theory*, **67** (2019), 3834–3841. https://doi.org/10.1109/TCOMM.2019.2900039

21. F. J. MacWilliams, A theorem on the distribution of weights in a systematic code, *Bell. Syst. Tech. J.*, **42** (1963), 79–94. https://doi.org/10.1002/j.1538-7305.1963.tb04003.x

22. F. J. MacWilliams, N. J. Sloane, *The theory of error-correcting codes*, North Holland, 1977.

23. R. J. McEliece, Weight congruence for *p*-ary cyclic codes, *Discrete Math.*, **3** (1972), 177–192. https://doi.org/10.1016/0012-365X(72)90032-5

24. M. Mondelli, *From polar to Reed-Muller codes*, Thesis, EPFL, 2016.

25. M. Mondelli, S. H. Hassani, R. L. Urbanke, From polar to Reed-Muller codes: A technique to improve the finite-length performance, *IEEE T. Commun.*, **62** (2014), 3084–3091. https://doi.org/10.1109/TCOMM.2014.2345069

26. S. Mesnager, A. Oblaukhov, Classification of the codewords of weights 16 and 18 of the Reed-Muller code RM (n-3, n), *IEEE T. Inform. Theory*, **68** (2021), 940–952. https://doi.org/10.1109/TIT.2021.3128495

27. D. E. Muller, Application of boolean algebra to switching circuit design and to error detection, *T. IRE Prof. Group Electron. Comput.*, **3** (1954), 6–12. https://doi.org/10.1109/IREPGELC.1954.6499441

28. V. S. Pless, W. C. Huffman, R. A. Brualdi, *Handbook of coding theory*, Elsevier, 1998.

29. I. S. Reed, A class of multiple-error-correcting codes and the decoding scheme, *T. IRE Prof. Group Electron. Comput.*, **4** (1954), 38–49. https://doi.org/10.1109/TIT.1954.1057465

30. M. Shi, X. Li, A. Neri, P. Solé, How many weights can a cyclic code have? *IEEE T. Inform. Theory*, **66** (2019), 1449–1459. https://doi.org/10.1109/TIT.2019.2946660

31. N. J. Sloane, *Online encyclopedia of integer sequences*, 1999. https://doi.org/10.1515/9780691197944-009

AIMS Press