Serdica J. Computing **13** (2019), No 1–2, 17–26

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

DISTRIBUTION OF THE BOOLEAN FUNCTIONS OF n VARIABLES ACCORDING TO THEIR ALGEBRAIC DEGREES*

Valentin Bakoev

Abstract. Knowledge about the enumeration and distribution of Boolean functions according to their algebraic degrees is important for the theory as well as for its applications. As of now, this knowledge is not complete: for example, it is well-known that half of all Boolean functions have a maximal algebraic degree. In the present paper, a formula for the number of all Boolean functions of n variables and algebraic degree $= k$ is derived. A direct consequence from it is the assertion (formulated already by Claude Carlet) that when $n \to \infty$, almost a half of all Boolean functions of n variables have an algebraic degree $= n-1$. The results obtained by this formula were used in creating the sequence A319511 in the OEIS. The discrete probability a random Boolean function to have a certain algebraic degree is defined and the corresponding distribution is computed, for $3 \leq n \leq 10$. Four applications are considered: at the design and analysis of efficient algorithms for exact

ACM Computing Classification System (1998): G.2.1, G.3, E.3.

Key words: Boolean function, cryptography, algebraic degree, enumeration, distribution.

*This work was partially supported by the Research Fund of the University of Veliko Tarnovo (Bulgaria) under contract FSD-31-340-14/26.03.2019. Some of the results were announced at the 8th International Conference on Algebraic Informatics (CAI 2019), Niš, Serbia.

or probabilistic computing the algebraic degree of Boolean functions; when checking four test files for representativeness; when creating benchmark files of Boolean functions.

1. Introduction. Boolean functions play an important role in coding theory, modern cryptography, digital circuit theory, etc. [8, 4, 5, 3]. Different types of their representations are used in these areas, for example by: the vectors of their functional values (called Truth Table (TT) vectors), algebraic normal forms (ANFs), numerical normal forms, etc. The algebraic degree of Boolean function is defined by its ANF and it is one of the most important cryptographic parameters. When it is higher, the Boolean function(s) involved in the design of block ciphers, pseudo-random numbers generators in stream ciphers, hash functions, etc., are more resistant against cryptographic attacks.

Enumeration and distribution of the Boolean functions satisfying desired cryptographic parameters take a great part of their research [7]. Knowledge about the distribution of the Boolean functions according to their algebraic degrees is important for the theory, as well as for the design and analysis of efficient algorithms for computing the algebraic degree (as those proposed in [6]), for generating test examples for such algorithms, etc. As of now, this knowledge is partial—there are some theoretical results about the enumeration of Boolean functions of n variables having certain algebraic degrees. A lot of them are derived by establishing relations between the TT vector's weight and the algebraic degree of the corresponding Boolean function. In general, it is well-known that [4, 3, 6]:

- A Boolean function of n variables has an algebraic degree $= n$ if and only if its TT vector has an odd weight. Hence half of all such functions have an algebraic degree $= n$.
- The number of affine Boolean functions of n variables (i. e., having algebraic degree at most 1) is 2^{n+1}. The TT vectors of all such functions (except both constant functions) have weight 2^{n-1}.

Counting of monomials of degrees at most r and the number of codewords for Reed-Muller codes is outlined in [4, p. 38]. Further, on p. 49, Claude Carlet notes: "When n tends to infinity, random Boolean functions have almost surely algebraic degrees at least $n-1$ since the number of Boolean functions of algebraic degrees at most $n-2$ equals $2^{\sum_{i=0}^{n-2}\binom{n}{i}} = 2^{2^n-n-1}$ and is negligible with respect to the number 2^{2^n} of all Boolean functions. But we shall see that the functions of algebraic degrees $n-1$ or n do not allow achieving some other characteristics

(balancedness, resiliency, ...)". This is one more reason to explore the whole distribution of Boolean functions according to their algebraic degrees.

The outline of the paper is as follows. The basic notions are given in Section 2. Section 3 starts with Theorem 1 which gives a formula for enumeration of Boolean functions of n variables in accordance with their algebraic degrees. This formula was used in creating the sequence A319511 [9]. The cited assertion of Carlet follows as a direct consequence of the theorem. It is derived in a different way in Corollary 1. In Section 4, four applications are discussed: at the design and analysis of efficient algorithms for exact or probabilistic computing the algebraic degree of Boolean functions; when checking four test files (with samples of Boolean functions) for representativeness; when creating of benchmark files of Boolean functions.

2. Basic notions. Let $\mathbb{F}_2 = \{0, 1\}$ be the *field of two elements* with both operations: $x \oplus y$ (sum modulo 2, XOR) and $x.y$ (multiplication, AND, denoted simply by xy), for $x, y \in \mathbb{F}_2$. $\mathbb{F}_2^n$ is the *n-dimensional vector space* over $\mathbb{F}_2$, containing all 2^n binary vectors. If $a = (a_1, a_2, \ldots, a_n) \in \mathbb{F}_2^n$, then $\bar{a} = \sum_{i=1}^{n} a_i.2^{n-i}$ denotes the natural number corresponding to a and $\bar{a}$ is called a *serial number* of the vector a. A (Hamming) *weight* of the same vector a is the natural number $wt(a) = \sum_{i=1}^{n} a_i$, i. e., it is the number of non-zero coordinates of a. A *Boolean function* of n variables is a mapping $f : \mathbb{F}_2^n \to \mathbb{F}_2$. So, if $x_1, x_2, \ldots, x_n$ denote the variables of f, it maps any binary input $x = (x_1, x_2, \ldots, x_n) \in \mathbb{F}_2^n$ to a single binary output $y = f(x) \in \mathbb{F}_2$. The set of all Boolean functions of n variables is denoted by $\mathcal{B}_n$.

Any Boolean function $f \in \mathcal{B}_n$ can be represented in an unique way by the vector of its functional values, called a *Truth Table* vector. It is denoted by $TT(f) = (f_0, f_1, \ldots f_{2^n-1})$, where $f_i = f(a_i)$ and a_i is the i-th lexicographic vector of $\mathbb{F}_2^n$, for $i = 0, 1, \ldots, 2^n - 1$. Since there are 2^n binary outputs corresponding to the vectors of $\mathbb{F}_2^n$, it follows that $|\mathcal{B}_n| = 2^{2^n}$.

The *Algebraic Normal Form* (ANF) is other unique representation of any $f \in \mathcal{B}_n$. It is a multivariate polynomial of the form

$$f(x_1, x_2, \ldots, x_n) = \bigoplus_{u \in \mathbb{F}_2^n} a_{\bar{u}}\, x^u\,. \tag{1}$$

Here $u = (u_1, u_2, \ldots, u_n) \in \mathbb{F}_2^n$, $a_{\bar{u}} \in \{0, 1\}$, and x^u means the monomial

$x_1^{u_1}x_2^{u_2}\ldots x_n^{u_n} = \prod_{i=1}^{n} x_i^{u_i}$, where $x_i^0 = 1$ and $x_i^1 = x_i$, for $i = 1, 2, \ldots, n$. The *degree* of the monomial x^u is equal to the number of variables in the product and it is denoted by $\deg(x^u)$. Obviously, $\deg(x^u) = wt(u)$. The *algebraic degree* of the Boolean function $f \in \mathcal{B}_n$ is the highest degree among all monomials in the ANF of f. It is denoted by $\deg(f)$. When the constant functions zero and one are considered as functions of n variables, they are denoted by $\tilde{0}_n$ and $\tilde{1}_n$, correspondingly. If $n = 0$, then $\tilde{0}_0 = 0$ and $\tilde{1}_0 = 1$, i. e., the Boolean values 0 and 1 are considered as Boolean function of 0 variables. Usually, the degree of $\tilde{0}_n$ is defined as $\deg(\tilde{0}_n) = -\infty$, but $\deg(\tilde{1}_n) = 0$, independently of the number of the variables.

Remark 1. When $f \in \mathcal{B}_n$ is given by its $TT(f)$, its ANF can be computed either by a well-known transformation (algorithm) defined by a special *transformation matrix* [1], or by a function called the *binary Möbius transform* [3], or by a simple *divide-and-conquer butterfly algorithm* [4, 7], etc. All these transformations are equivalent, and depending on the area of consideration they are known as *ANF Transform (ANFT), fast Möbius (or Moebius) Transform, Zhegalkin Transform, Positive Polarity Reed-Muller Transform*, etc. [1]. For any $f \in \mathcal{B}_n$ there exists a unique ANF of it—as Zhegalkin's famous theorem states, or as is shown in [3, 4]. So the ANFT is a bijection between the functions in $\mathcal{B}_n$ and the set of their ANFs. Furthermore, the ANFT coincides with its inverse transformation and so it is an involution [1, 3, 4, 7].

3. Enumeration and distribution of the Boolean functions according to their algebraic degrees. Let $d(n,k)$ denotes the number of Boolean functions of $f \in \mathcal{B}_n$ such that $\deg(f) = k$.

Theorem 1. *For any integers $n \geq 0$ and $0 \leq k \leq n$, the number*

$$d(n,k) = \begin{cases} 1, & \text{if } k = 0; \\ (2^{\binom{n}{k}} - 1).2^{\sum_{i=0}^{k-1}\binom{n}{i}}, & \text{if } 1 \leq k \leq n. \end{cases} \tag{2}$$

Proof.

a) The particular case $k = 0$ means that $f = \tilde{1}_n = 1$, where 1 is considered as a unique monomial that contains no variables. In this case, the assertion of the theorem is true.

b) Let $1 \leq k \leq n$ and $X = \{x_1, x_2, \ldots, x_n\}$ be a set of variables. The set of monomials in any ANF (i. e., formula of the type (1)) of n variables can be

partitioned into two types subsets denoted by A and B. A subset of the type A contains all monomials of degree $= k$ and so $A \neq \emptyset$. A subset of the type B contains all monomials of degrees less than k and the case $B = \emptyset$ is possible. We shall enumerate all possible subsets of the type A and B, correspondingly:

1. There are $\binom{n}{k}$ ways to choose k variables from X and to form a monomial of degree $= k$. So, if M is the set of all such monomials, then $|M| = \binom{n}{k}$. There are $2^{\binom{n}{k}} - 1$ ways for at least one of them to be chosen and to form the subset A in the ANF, since so many are the elements of the power set of M without the empty set. In conclusion, the first multiplier in formula (2) represents the number of all possible combinations of (at least one) monomials having degree $= k$ in the ANFs, which is the number of all possible subsets of the type A.

2. Analogously, we have $\binom{n}{i}$ monomials of degree $= i$, for $i = 0, 1, \ldots, k-1$. Thus the set of all of them contains $\sum_{i=0}^{k-1} \binom{n}{i}$ monomials. Its power set has a cardinality of $2^{\sum_{i=0}^{k-1} \binom{n}{i}}$ which is the number of all possible subsets of the type B. This number includes the case $B = \emptyset$ which means that no monomial of degree $< k$ is chosen and so the corresponding ANFs contain only monomials of degree $= k$. In conclusion, the second multiplier in formula (2) is the number of all possible subsets of the type B.

Finally, any subset of the type A can be combined with any subset of the type B in the formulas of the type (1), which is the reason for applying the multiplication rule between both terms in formula (2). □

Table 1 represents the values of $d(n, k)$ obtained by formula (2), for $n = 0, 1, \ldots, 5$ and $0 \leq k \leq n$.

Remark 2. For any positive natural number n, we can define the relation "*equal algebraic degrees*" over the set $\mathcal{B}_n$ as follows: arbitrary functions $f, g \in \mathcal{B}_n$ belong to this relation iff $\deg(f) = \deg(g)$. It is easy to verify that it is an equivalence relation. Therefore this relation partitions the set $\mathcal{B}_n$ into $n + 2$ equivalence classes, each of which contains all Boolean functions of equal degrees. The first of them contains only the constant zero function, and the cardinalities of the remaining classes can be obtained by Theorem 1. Table 1 shows these cardinalities for $0 \leq n \leq 5$.

Table 1. The values of $d(n,k)$, for $n = 0, 1, \ldots, 5$ and $0 \leq k \leq n$

	The values of $d(n,k)$, for:					
$n =$	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
0	1					
1	1	2				
2	1	6	8			
3	1	14	112	128		
4	1	30	2016	30720	32768	
5	1	62	65472	67043328	2080374784	2147483648

More comments, relations and results about the numbers $d(n,k)$ (for $n \leq 10$) can be seen in [9], sequence A319511. They suggest the following assertion.

Corollary 1. *The number* $d(n, n-1)$ *tends to* $\frac{1}{2} \cdot |\mathcal{B}_n|$ *when* $n \to \infty$.

Proof. Applying formula (2) for $k = n-1$, we compute the limit of the ratio between $d(n, n-1)$ and the number of all Boolean functions in $\mathcal{B}_n$ as follows:

$$\lim_{n\to\infty} \frac{d(n,n-1)}{|\mathcal{B}_n|} = \lim_{n\to\infty} \frac{(2^{\binom{n}{n-1}} - 1).2^{\sum_{i=0}^{n-2}\binom{n}{i}}}{2^{2^n}} = \lim_{n\to\infty} \frac{(2^n - 1).2^{2^n-n-1}}{2^{2^n}} =$$

$$\lim_{n\to\infty} \frac{2^{2^n-1} - 2^{2^n-n-1}}{2^{2^n}} = \lim_{n\to\infty} \left(\frac{1}{2} - \frac{1}{2^{n+1}}\right) = \frac{1}{2}.$$

□

Obviously, when n grows and k becomes close to n, the values of $d(n,k)$ grow extremely fast. It is convenient to define and use the discrete probability $p(n,k)$ a random Boolean function $f \in \mathcal{B}_n$ to have an algebraic degree $= k$,

$$p(n,k) = \frac{d(n,k)}{|\mathcal{B}_n|} = \frac{d(n,k)}{2^{2^n}},$$

for $n \geq 0$ and $0 \leq k \leq n$. The values of $p(n,k)$ obtained for a fixed n give the distribution of the functions from $\mathcal{B}_n$ in accordance with their algebraic degrees. Table 2 represents partially[1] this distribution, for $3 \leq n \leq 10$ and $n-3 \leq k \leq n$. The values of $p(n,k)$ in it are rounded up to 10 digits after the decimal point.

4. Applications. The results obtained here have some useful applications:

[1] Because the remaining values are rounded to zero.

Table 2. Partial distribution of the function from $\mathcal{B}_n$ according to their algebraic degrees, for $n = 3, 4, \ldots, 10$

n	The values of $p(n,k)$, for:			
	$k = n-3$	$k = n-2$	$k = n-1$	$k = n$
3	0.00390625	0.0546875	0.4375	0.5
4	0.0004577637	0.0307617187	0.46875	0.5
5	0.0000152439	0.0156097412	0.484375	0.5
6	0.0000002384	0.0078122616	0.4921875	0.5
7	0.0000000019	0.0039062481	0.49609375	0.5
8	0	0.0019531250	0.498046875	0.5
9	0	0.0009765625	0.4990234375	0.5
10	0	0.0004882812	0.4995117187	0.5

1. The knowledge about the distribution of the function from $\mathcal{B}_n$ according to their algebraic degrees is very useful in the design and analysis of algorithms for efficient computing the algebraic degree of Boolean functions. Such algorithms are considered in [6, 2]. The continuations outlined in [2] (presented at the conference CAI 2019, but still unpublished) demonstrate the benefit of this knowledge.

2. A fast and very simple *probabilistic algorithm* for fast computing the algebraic degree of $f \in \mathcal{B}_n$, given by its $TT(f)$. It can be described in short as: "Compute the weight of $TT(f)$. If it is an odd number, return n, else, return $n-1$." From Theorem 1, Corollary 1 and Table 2, it follows that for arbitrary $f \in \mathcal{B}_n$ this algorithm will return a correct output in $\approx 100\%$ of all such functions. In addition, the probability of "*the algorithm returns a correct output*" tends to 1 when $n \to \infty$.

   In the continuations of [2], we showed that if the $TT(f)$ has a byte-wise representation, an algorithm for computing the weight of $TT(f)$ will have a time complexity $\Theta(2^n)$. When the $TT(f)$ has a bitwise representation in $64 = 2^6$-bit computer words, it occupies $s = 2^{n-6}$ such words. Thus, by using a look-up table – an array a of size 2^{16} elements which are precomputed weights of integers (i. e., $a[i] = weight(i)$, for $i = 0, 1, \ldots, 2^{16} - 1$) – the weight of $TT(f)$ will be computed in $\Theta(4.s) = \Theta(2^{n-4})$ steps. But we realized that we need the parity check of $TT(f)$ instead of its weight. The parity check of $TT(f)$ can be computed significantly more efficiently—in $\Theta(s-1+6) = \Theta(2^{n-6})$ steps. The last comments show how fast can be the probabilistic algorithm and what improvement to expect in the efficiency of exact algorithms when they use a bitwise representation of $TT(f)$ and

bitwise operations.

3. It is known that "With today's computation power, it is not possible to conduct an exhaustive search on Boolean functions of 6 variables or more" [7]. That is why the algorithms for investigation of such Boolean functions use samples of them which should be representative. The results obtained here were used to check four files for representativeness. These files contain 10^6, 10^7, 10^8 and 10^9 randomly generated unsigned integers in 64-bits computer words. These integers were obtained by using the usual `rand ()` function in C/C++ programming language, without any additional checks and conditions. The files were used as test examples in [1, 2]. When Boolean functions of $n \geq 6$ variables are used, 2^{n-6} integers are read from the selected file and so they form the $TT(f)$ (in a bitwise representation) of the serial Boolean function. We used each of these files as an input of Boolean functions of $n = 6, 8, 10, 12, 14, 16$ variables. We computed their ANFs, thereafter the algebraic degrees of each of these functions, and we enumerated all functions of equal degrees. Finally, we compute the deviations – the absolute values of the differences between the theoretical and computed distributions. The obtained results are given in Table 3.

Table 3. Experimental results about the test files representativeness— the maximal deviations in percents

Number of integers:	Maximal deviations in % for Boolean functions of:					
	6 vars	8 vars	10 vars	12 vars	14 vars	16 vars
10^6	0.64775	0.018	0.1504	0.0922	0.8479	1.2303
10^7	0.62853	0.04393	0.10429	0.17216	0.29952	0.10753
10^8	0.623065	0.007604	0.027972	0.031936	0.003222	0.261884
10^9	0.623232	0.0079967	0.0099744	0.014511	0.01743	0.092923

We have to note that the two test files of smaller sizes have been used in the development and debugging of the algorithms in [1, 2], whereas the two test files of larger sizes have been used for the true tests and results. So we consider that the algorithms work with samples of Boolean functions which are representative enough and their results are real.

4. The recent results can be used in creating benchmarks files containing samples of Boolean functions whose distribution (according to their algebraic degrees) will be as close as necessary to the theoretical distribution. Thus we can obtain significantly more representative samples than those in the test files mentioned above.

5. Conclusions. The main goal of this paper is to contribute knowledge about the enumeration and distribution of Boolean functions according to their algebraic degrees, as well as to make the knowledge of this matter more popular. We demonstrated four applications of it as illustrations of its use. We hope this knowledge will also find other applications.

REFERENCES

[1] Bakoev V. Fast Bitwise Implementation of the Algebraic Normal Form Transform. *Serdica Journal of Computing*, **11** (2017), No 1, 45–57.

[2] Bakoev V. Fast Computing the Algebraic Degree of Boolean Functions. In: M. Ćirić, M. Droste, J-É. Pin (eds). Algebraic Informatics. 8th International Conference, CAI 2019, Niš, Serbia. *Lecture Notes in Computer Science*, **11545** (2019), 50–63.

[3] Canteaut A. Lecture notes on Cryptographic Boolean Functions. Inria, Paris, France, 2016.

[4] Carlet C. Boolean Functions for Cryptography and Error Correcting Codes. In: Y. Crama and P. L. Hammer (eds). Boolean Models and Methods in Mathematics, Computer Science, and Engineering. Cambridge Univ. Press, 2010, 257–397.

[5] Carlet C. Vectorial Boolean Functions for Cryptography. In: Y. Crama and P. L. Hammer (eds). Boolean Models and Methods in Mathematics, Computer Science, and Engineering. Cambridge Univ. Press, 2010, 398–469.

[6] Climent J.-J., García F., Requena V. The degree of a Boolean function and some algebraic properties of its support. In: Data Management and Security. WIT Press, 2013, 25–36.

[7] Çalik Ç. Computing Cryptographic Properties of Boolean Functions from the Algebraic Normal Form Representation. Ph. D. Thesis, Middle East Technical University, Ankara, Turkey, 2013.

[8] MacWilliams F. J., Sloane N. J. A. The Theory of Error-Correcting Codes. Amsterdam, North-Holland, 1978.

[9] OEIS Foundation Inc. The On-line Encyclopedia of Integer Sequences. https://oeis.org/, 31 January 2020.

Valentin Bakoev
Faculty of Mathematics and Informatics
St. Cyril and St. Methodius University
Veliko Tarnovo, Bulgaria
e-mail: v.bakoev@ts.uni-vt.bg

Received March 6, 2019
Final Accepted July 9, 2019