PUBLICATIONS DE L'INSTITUT MATHÉMATIQUE
Nouvelle série tome 42 (56), 1987, pp. 143–147

SOME PROPERTIES OF THE COMBINATIONAL MEASURE OF COMPLEXITY OF BINARY WORDS

S. Stojanović and **B. Vidaković**

Abstract. We state and prove some basic properties of the measure $C(x)$ introduced in Vidaković [4], and draw a parallel between this and known ineffective measures of complexity.

Introduction

The theory of algorithmical complexity of finite binary words that has been rather developed since the pioneering works of Kolmogorov [I] and Chaitin [2] mainly deals with ineffective measures of complexity. The best one can do is to approximate an ineffective measure by a general recursive function, but the approximation would not be constructive. The combinational complexity of a binary word f of length 2^n is defined by combinational complexity of the Boolean function represented by the word x. The measure introduced this way is effective and it can be extended to words of arbitrary length. Many of its features are similar to those of the known ineffective measures.

Notation and definitions

Let us suppose the given alphabet is $A = \{0,1\}$, and X_n to be the set of all words of length n; $X = \cup_n X_n$ will denote the set of all finite words and x, y, z elements of X. $l(x)$ is the length of the word x. To record a pair of words $x = x_1 x_2 \dots x_n$ and y in a form of one word, we use the coding $\overline{x}y$ where $\overline{x} = x_1 x_1 x_2 x_2 \dots x_n x_n 01$. By a^k we denote the product of concatenating a k times. By $F(X) \preccurlyeq G(X)$ we denote the predicate $(\exists C)(\forall x) F(x) \leq G(x) + C$.

The complexity of the word x with respect to a partially recursive function F is the number $K_F(x) = \min\{l(p) \mid F(p) = X\}$. There exists an optimal partially recursive function F^0 such that for any other partially recursive function G and for any x one has $K_{F0}(x) \leq K_G(x) + C$, where C depends only on G. Instead of $K_{F0}(x)$ we write $K(x)$; the basic properties of this measure can be found in [6].

AMS Subject Classification (1980): Primary 60A04, 68C25.

Let $\mathcal{F}_n$ be the set of all Boolean functions of n variables and &, $\vee$, $\neg$ be respectively the conjunction, disjunction and negation symbols. They make the base $\mathcal{B}_0$. Let further Γ be an oriented acyclic graph. We shall say that a vertex of Γ is of type (p,q) if the input degree of that vertex is p and the output degree is q. We shall consider only graphs all of whose vertices are of type $(0,1)$ (inputs), $(1,1)$ (negation), $(2,1)$ (conjunction and disjunction), $(1,p)$, $p \geq 2$ (branching), or $(p,0)$, $p = 1,2$ (outputs); see fig. 1.

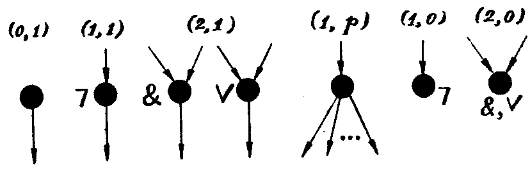

Fig. 1

Vertices of type (1,0), (1,1), (2,0) or (2,1) will be called interior and each of them realizes a Boolean function of n arguments. We shall say that the graph Γ realizes f if one of its inside vertices realizes f. The graph Γ obtained from the function f by a given synthesis algorithm $\mathcal{A}$ will be denoted $\Gamma_{f,\mathcal{A}}$. Let v be the lexicographical coding $v : \{1, 2, \ldots, 2^n\} \to X_{2^n}$. For each word $x = x_1x_2, \ldots, x_{2^n} \in X_{2^n}$ there is a unique function $f \in \mathcal{F}_n$ such that $f(v(i)) = x_i$, $i = 1, 2, \ldots, 2^n$. For that reason in the sequel we shall not distinguish between words of length 2^n and corresponding Boolean functions. If $|\Gamma|$ denotes the number of interior vertices of the graph Γ then $L_{\mathcal{A}}(x) = |\Gamma_{x,\mathcal{A}}|$ is the combinational complexity of the Boolean function x with respect to the synthesis algorithm $\mathcal{A}$.

Definition 1. [4] Combinational complexity of the word $x \in X_{2^n}$ with respect to the synthesis algorithm $\mathcal{A}$ is the number $C_{\mathcal{A}}(x) = nL_{\mathcal{A}}(x)$.

Remark. The multiplier n is necessary to ensure the introduced measure of complexity be in accordance with the already existing measures.

The apparent difficulty that the introduced measure depends on the choice of the synthesis algorithm is disposed by the following

THEOREM 1. *There exists an algorithm $\mathcal{A}_0$ (the searching algorithm) such that for any other algorithm $\mathcal{B}$ and $x \in X_{2^n}$ one has*

$$C_{\mathcal{A}_0}(x) \leq C_{\mathcal{B}}(x)$$

Proof. There is only finitely many acyclic graphs with n inputs and a fixed number of interior vertices. Also it is possible to check if some of their interior vertices realize x. Since the base $\mathcal{B}_0$ is complete this searching procedure eventually terminates. □

We shall denote $D_{\mathcal{A}_0}(x)$ simply by $C(x)$.

Remark. The function $C(x)$ is a constructive function, but even for small lengths of words, for example for $n = 4$ $(l(x) = 2^4)$ we have to check a large number of graphs. For $n = 5$ it is still possible to solve the problem with a computer (it takes a few hours of machine working time). But for $n > 8$ the effective calculation of the function value $C(x)$ is practically impossible.

The basic properties of the measure $C(x)$

THEOREM 2. *For "almost all" words $x \in X_{2^n}$ one has $C(x) > 2^n$.*

Proof. Let $N(n,m)$ be the number of minimal graphs of complexity m with n inputs. It can be shown that $N(n,m) \le (n+m)^{2m}3^m/m!$. Each minimal graph corresponds to only one function. Indeed if a graph is minimal for two function f_1 and f_2, then by elimination of the vertex realizing f_1 we obtain a graph which, due to acyclicity, realizes f_2, contradicting the minimality hypothesis. Let us now establish a correspondence between minimal schemes and words in the alphabet $A \times A \times \mathcal{B}_0$, where $A = \{x_1, \ldots, x_n, 1, \ldots, m\}$, $x_1, x_2, \ldots, x_n$ are inputs, $\mathcal{B}_0 = \{\neg, \&, \wedge\}$, and $1, 2, \ldots, m$ are codes of interior vertices. If the element i is preceded by elements a and b (We assume $a = b$ in the case of $\neg$) whose codes are v_a and v_b and if the element i is of type $B_i \in \mathcal{B}_0$ than we define the i-th letter of the word s coding our graph to be (v_a, v_b, B_i). From the word s it is possible to reconstruct the graph up to isomorphism. The number of different words of length m in the alphabet $A \times A \times \mathcal{B}_0$ is $3^m(n+m)^m(n+m)^m$. The $m!$ different codings of interior vertices all give rise to isomorphic schemes, so we have $N(m,n) = 3^m(n+m)^{2m}/m!$. Let $\overline{N}(m,n) = \sum_{i=1}^{m} N(i,n)$; then $\overline{N}(m,n) \le (m+n)^{2m+2}/(m+n-1)^2$. Since $\ln(\overline{N}(2^n/n,n)/2^{2^n}) \to -\infty$, we have $\overline{N}(2^n/n,n) = 2^{2^n} \to 0$, as $n \to \infty$. In other words, the relative frequency of words from X_{2^n} whose complexity does not exceed 2^n converges to 0, which proves the theorem. □

Let $w(x) = \sum_{i=1}^{l(x)} x_i$ be the "weight" of the word x. It can be proved that "almost all" words x have weight close to $l(x)/2$, i.e. $|w(x) - 2^{n-1}| \le n2^{n/2}$. Any considerable deviation of the weight from half-length should decrease the complexity. Indeed

THEOREM 3. *If $x \in X_{2^n}$, then $C(x \mid w(x) = s) \le n^2(\min\{s, 2^n - s\} + 1)$.*

Proof. Let, $\mathcal{A}$ be the algorithm of synthesis of PDNF. Let $\varGamma_k$ be the graph realizing the elementary conjunction $K = x_1^{\sigma_1}\&\ldots\&x_n^{\sigma_n}$, where $\sigma_i \in \{0,1\}$ and

$$x_i^{\sigma_i} = \begin{cases} x_i, & \sigma_i = 1, \\ \neg x_i, & \sigma_i = 0. \end{cases}$$

Then $|\varGamma_k| = n + (n-1)$, The weight s of the word x is the number of elementary conjunctions, so $|\varGamma_{x,\mathcal{A}}| \le n + s(n-1) + s - 1 \le n(s+1)$. This estimation should be taken for words of small weight. An analogous estimation applies in the case of big weights we take PCNF which consists of $2^n - s$ elementary disjunctions. □

We note that for every n there exist words $x \in X_{2^n}$ such that $C(x) = 0$. A typical example is $111\ldots1000\ldots0$; the corresponding graph has no interior vertices and values are "read" off from the input x_1.

If we order the set X_{2^n} respecting the increasing of complexity and denote $C^*(2^n) = \max\{C(x), x \in X_{2^n}\}$, then the so called "Shannon effect" holds for the measure $C(x)$, i.e.

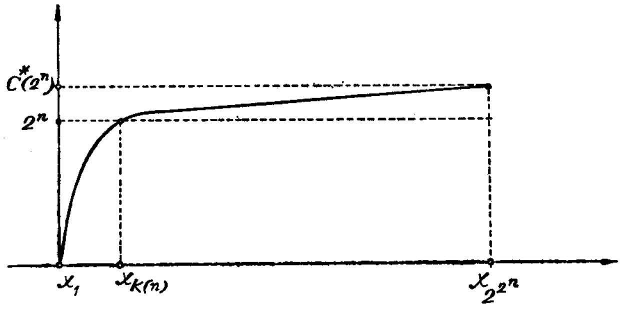

Fig. 2

$k(n)/2^{2^n} \to 0$ (Fig. 2, Theorem 2). This effect shows that the measure C is correctly defined, i.e. that non-complex words are few in number and that the complexity of most words is close to their length.

The presence of any definable regularity decreases the measure of complexity. For example:

(i) If $x = 0^{2^n}$ or $x = 1^{2^n}$, then $C(x) \leq 2n$.

(ii) If the word $x \in X_{2^n}$ corresponds to a symmetric function, then $C(x) \leq c \cdot n^2$, where c depends on the choice of the base. (For bases $\mathcal{B}_0$ and $\mathcal{F}_2$ one has respectively $c = 52$ and $c = 5$.)

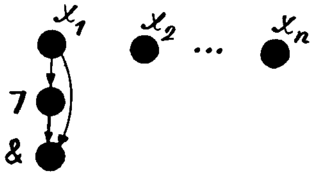

Fig. 3

(iii) If the word $x \in X_{2^n}$ corresponds to a symmetric function, then $\#\{x \,|\, C(x) \geq 5n^2/2 - 5n\} \geq 2^n$.

The statements (ii) and (iii) are proved in [3]. For (i) we have on Fig. 3. depicted a scheme realizing $x = 0^{2^n}$. Since $|\Gamma_x| = 2$ holds, $C(x) \leq 2n$. The case $x = 1^{2^n}$ is analogous.

THEOREM 4. *For every word* $X \in X_{2^n}$ *one has* $K(x) \preccurlyeq C(x)(2+O(l(n)/n))$.

Proof. Let a word p and a function F be such that F applied to p synthetises a scheme Γ_x and then x itself. All symbols of the alphabet A (Theorem 2.) are codes by words of length $l(n+L(x))$, which we denote t. As a code for x one can take

$$p=\overline{t}|a_1'a_1''s_1|a_2'a_2''s_2|\dots|a_{L(x)}'a_{L(x)}''s_{L(x)}|,$$

where $a_i', a_i'' \in A$, $s_i \in \mathcal{B}_0$, $i=1,2,\dots,L(x)$. So

$$\begin{aligned} K(x) &\preccurlyeq K_F(x) \le l(p) = 2l(t)+2+L(x)(2l(a)+l(s)) \le \\ &\le n^{-1}C(x)(2l(C(x)/n+n)+2)+2l(l(C(x)/n)+2 \\ &\le C(x)(2+O(l(n)/n)). \quad \square \end{aligned}$$

THEOREM 5. *For every word $X \in X_{2^n}$ one has $K(C(x)) \preccurlyeq K(x)$.*

Proof. Let $F_0(p_x)=x$, i.e. $K(x)=l(p_x)$. Let G be the function which calculates x using the program p_x, and then finds a minimal scheme Γ_x and finaly calculates $C(x)$. We have

$$K(C(x)) \preccurlyeq K_G(C(x)) = l(p_x) = K(x). \quad \square$$

It is of practical interest to have the measure of randomness of binary words which is easily calculable. It is possible to define nonparametric tests of randomness of binary words based on the their measure of complexity. The word x is random if $C(x)$ is close to $l(x)$, that is if

$$C(x) \ge l(x) - A(\alpha, l(x)),$$

where $A(\alpha, l(x))$ is the constant depending on the given significance level α and the length $l(x)$.

REFERENCES

[1] А. Н. Колмогоров, *Три подхода к определении понятия "количество информации"*, Пр. Пер. Инф. **1** (1965) 3–7.

[2] G. J. Chaitin, *On the length of programs for computing finite binary sequences*, J. ACM 13, **4** (1966), 547–569.

[3] L. J. Stockmeyer, *On the combinational complexity of certain symetric Boolean functions*, Math. Syst. Th. **10** (1977), 323–336.

[4] B. Vidaković, *Jedna efektivna mera složenosti binarnih reči*, Mat. Vesnik **37** (1985), 327–332.

[5] Р. Г. Нигматуллин, *Сложность Булевых функций*, Казан. Унив., 1983.

[6] А. К. Звонкин, Л. А. Левин, *Сложность конечных обьектов и обоснование поныатия информации и случайности с помощью теории алгоритмов*, УМН **25**, 6 (1956), 1970, 85–127.

[7] D. E. Knuth, *The Art of Computer Programming*, *Vol.* 2, *Seminumerical Algorithms*, Addison-Wesley, 1969.

Institut za matematiku, PMF
Studentski trg 16
11000 Beograd
Jugoslavija

Purdue University
Department of Statistics
West Lafayette, Indiana 47907
USA

(Received 05 03 1987)