ISSN 1066-369X, Russian Mathematics (Iz. VUZ), 2014, Vol. 58, No. 5, pp. 30–42. © Allerton Press, Inc., 2014.
Original Russian Text © I.S. Sergeev, 2014, published in Izvestiya Vysshikh Uchebnykh Zavedenii. Matematika, 2014, No. 5, pp. 38–52.

Upper Bounds for the Formula Size of Symmetric Boolean Functions

I. S. Sergeev[1]*

[1]*Moscow State University, GSP-1, Leninskie Gory, Moscow, 119991 Russia*

Received November 7, 2012

Abstract—We prove that the complexity of the implementation of the counting function of n Boolean variables by binary formulas is at most $n^{3.03}$, and it is at most $n^{4.47}$ for DeMorgan formulas. Hence, the same bounds are valid for the formula size of any threshold symmetric function of n variables, particularly, for the majority function. The following bounds are proved for the formula size of any symmetric Boolean function of n variables: $n^{3.04}$ for binary formulas and $n^{4.48}$ for DeMorgan ones. The proof is based on the modular arithmetic.

DOI: 10.3103/S1066369X14050041

Keywords: *formula size, symmetric Boolean functions, majority function, multiplication.*

1. INTRODUCTION

In this paper we study the complexity of the implementation of symmetric Boolean functions by formulas over the basis B_2 of all binary Boolean functions and over the standard basis $B_0 = \{\wedge, \vee, ^-\}$. (A function is called symmetric, if its values are independent of permutations of arguments; in the Boolean case this property is equivalent to the fact that values of the function depend on the arithmetic sum of arguments, only.)

Recall that any formula over the basis B, the formula size, the formula depth, and the function implemented by the formula, are inductively defined as follows:

0) symbols of variables are formulas of size 1 and depth 0; they implement the corresponding identity functions;

1) the expression $G(F_1, \dots, F_k)$, where the symbol G denotes a k-place function $g \in B$, and F_i is a formula of size L_i and depth D_i implementing a function f_i, is a formula of size $L_1 + \dots + L_k$ and depth $\max\{D_1, \dots, D_k\} + 1$ implementing the function $g(f_1, \dots, f_k)$.[1)]

If a basis B consists of no more than binary Boolean functions, then we can use the following compact notation of the formula: $(F_1 \circ F_2)$, where $\circ$ denotes a binary operation over the basis B, or $\overline{F_1}$ in the case of the unary negation operation. Here we can remove parentheses, when the priority of operations is definite or insignificant.

The complexity $L_B(f)$ of the implementation of a Boolean function f by formulas over the basis B is defined as the minimum complexity of formulas that implement f. The complexity $L_B(K)$ of the class (the set) of functions K is defined as $\max_{f \in K} L_B(f)$. The formula implementing a Boolean operator is defined as the totality of formulas implementing separate functions that represent components of the operator. The complexity of a Boolean operator is defined as the sum of complexities of its components. See [1–6] for a more detailed description of the introduced notions (as well as the notion of the Boolean circuit; we use it below in this paper, but it is not essential for stating the main results).

*E-mail: isserg@gmail.com.

[1)]One can define the size and depth of a formula in another way; for example, define the size as the number (the sum of weights) of basic functions used for constructing the formula, and neglect unary functions when calculating the depth. Results obtained in this paper remain also valid for the mentioned variants of definitions.