Olga V. Podolskaya

Circuit complexity of symmetric Boolean functions in antichain basis

DOI: 10.1515/dma-2016-0003
Received February 16, 2015

Abstract: We study the circuit complexity of Boolean functions in an infinite basis consisting of all characteristic functions of antichains over the Boolean cube. For an arbitrary symmetric function we obtain the exact value of its circuit complexity in this basis. In particular, we prove that the circuit complexities of the parity function and the majority function of n variables for all integers $n \geqslant 1$ in this basis are $\lfloor \frac{n+1}{2} \rfloor$ and $\left\lfloor \frac{n}{2} \right\rfloor + 1$ respectively. For the maximum circuit complexity of n-variable Boolean functions in this basis, we show that its order of growth is equal to n.
The research is supported by the Russian Foundation for Basic Research, project 14–01–00598.

Keywords: Boolean circuit complexity, antichain functions, Boolean circuits, symmetric Boolean functions, the Shannon function

Note: Originally published in *Diskretnaya Matematika* (2015) **27**, №3, 95–107 (in Russian).

1 Introduction

A Boolean function is called *symmetric* if any permutation of its variables does not change the value of the function. Any n-variable symmetric function f is determined by a binary tuple $(a_0, a_1, \ldots, a_n)$, where $a_i = 1$ iff the function f equals 1 on inputs with i ones. For a function f let us denote by $k(f)$ the number of indices i such that a_i is equal to 1. From now on by symmetric functions we mean symmetric Boolean functions.

The Boolean function $p_n(x_1, \ldots, x_n) = x_1 + \ldots + x_n \pmod 2$ is called the *parity function*. The Boolean function $m_n(x_1, \ldots, x_n)$ such that it equals to 1 iff the number of ones in an input is at least $n/2$ is called the *majority function*.

Consider the Boolean cube $\{0,1\}^n$ as a partially ordered set of tuples with the natural order of the Cartesian product ($0 \leqslant 1$). An *antichain* is a subset of the Boolean cube such that no two of its tuples are comparable. An *antichain function* is a function that takes value 1 only on a certain antichain over the Boolean cube. We denote by AC the set of all antichain functions [1]. This set is closed under two operations: substitution of constants and identification of variables, and any Boolean function may be expressed via functions from the set AC and the operation of superposition.

Let two real functions $a(n)$ and $b(n)$ of natural n take positive values for all sufficiently large n. We say that the order of growth of function $a(n)$ is less than $b(n)$ and denote this by $a(n) = O(b(n))$ if there exists a constant $c > 0$ such that $a(n) \leqslant cb(n)$ for all sufficiently large n. At the same time we say that the order of growth of function $b(n)$ is at least $a(n)$ and denote this by $b(n) = \Omega(a(n))$. If simultaneously $a(n) = \Omega(b(n))$ and $a(n) = O(b(n))$, then we say that the order of growth of $a(n)$ equals $b(n)$ and denote this by $a(n) = \Theta(b(n))$.

For the infinite AC basis we study circuits computing symmetric functions. The definition of the circuit and other notations used in the paper may be found, e.g., in [4].

Let *circuit complexity* denote the number of gates in the circuit. Let *complexity of a function* denote the minimum number of gates in the circuit computing this function. Let $L(S)$ denote the complexity of a cir-

Olga V. Podolskaya: Lomonosov Moscow State University, e-mail: olgavikonov@gmail.com

cuit S and $L(f)$ denote the complexity of a function f. Let the *Shannon function* denote the function $L(n) = \max L(f)$, where maximum is taken over all n-variable Boolean functions f.

Lower bounds on circuit complexity in the AC basis have been studied in [1], [2], [5]. In [1] O. M. Kasim-Zade had proved the bound $\Omega(n^{1/3})$ for the complexity of the n-variable parity function. This result leads to the lower bound for the Shannon function: $L(n) = \Omega(n^{1/3})$. Later in [2] O. M. Kasim-Zade proved the lower bound $\Omega((n/\ln n)^{1/2})$ for the parity function and the Shannon function. By the extension of the method used in [1], the latter result was improved in [5] where the lower bound $\Omega(\sqrt{n})$ was obtained for the parity function, the majority function and almost all Boolean functions of n variables. That is, the lower bound $L(n) = \Omega(\sqrt{n})$ for the Shannon function was established in the AC basis.

In Section 3 for any symmetric function we obtain the exact value of the circuit complexity in the AC basis.

Theorem 1. *For all natural n and for any symmetric function $f(x_1 \dots, x_n)$*

$$L(f) = \min(k(f),\ n - k(f) + 2).$$

By means of Theorem 1 it is easy to obtain the exact value of the circuit complexity in the AC basis for the n-variable parity function and majority function.

Theorem 2. *For all natural n*

$$L(p_n) = \left\lfloor \frac{n+1}{2} \right\rfloor,\ L(m_n) = \left\lfloor \frac{n}{2} \right\rfloor + 1.$$

This theorem is proved in Section 4.

As to the upper bounds, the estimate $L(n) \leqslant n + 1$ for the Shannon function was proved in [1] for all natural n. In [6] the stronger upper bound $L(n) \leqslant n$ was stated and the main ideas of the proof were shown. The latter bound along with the lower bound following from Theorem 2 give the order of growth of the Shannon function in the AC basis.

Theorem 3. *In the AC basis $L(n) = \Theta(n)$.*

2 Preliminaries

Denote by $[n]$ the set of natural numbers $\{1, 2, \dots, n\}$.

Denote by $\boldsymbol{x}$ the tuple of arbitrary values of variables $(x_1, \dots, x_n)$.The maximal Boolean n-tuple $\mathbf{1} = (1, 1, \dots, 1)$ is called the *top* tuple of the Boolean cube, and the minimal Boolean n-tuple $\mathbf{0} = (0, 0, \dots, 0)$ is called the *bottom* one. Similarly, for any subcube of a lower dimension its maximal and minimal tuples are called the top and the bottom ones respectively.

For any $P \subseteq [n]$, denote by $\boldsymbol{x}^P$ a binary tuple $\boldsymbol{x} = (x_1, \dots, x_n)$ such that $x_p = 1$ for $p \in [n]$ iff $p \in P$.

The *layer* of the Boolean cube *with the index t*, where $t = 0, 1, \dots, n$, we define as the set of all n-tuples containing exactly t ones. Clearly, any such layer is the antichain in the Boolean cube. In these terms, the support of a symmetric function f that equals 1 only on inputs with $i_1, \dots, i_{k(f)}$ ones is a union of the layers with the indices $i_1, \dots, i_{k(f)}$ respectively.

We call a circuit *reduced* if the different inputs of any gate are connected with different gates of the circuit and each gate computes a function other than the constant (see [2]). Note that for any circuit in the AC basis of complexity s computing a function other than the constant there exists a reduced circuit of complexity at most s in the AC basis computing the same function.

We call a numeration on circuit gates *regular* if each input of each circuit gate is connected to an output of a gate with the smaller index or to a circuit input. It is known that a regular numeration may be introduced on any circuit (possibly, by several ways) [3].

From now on in the proofs of lower bounds we consider reduced circuits in the AC basis with fixed arbitrary regular numerations on their gates.

Denote by e_l the gate with the index l, by g_l the antichain function corresponding to this gate and by h_l the function computed on the output of the gate e_l. For a given input $\boldsymbol{x}$ denote by $h_k(\boldsymbol{x})$ the value of the function h_k. By *input variables* of a circuit we call the symbols of variables $x_1, x_2, \ldots, x_n$ assigned respectively to n inputs of the circuit.

3 Proof of Theorem 1

Let us prove two lemmas.

Lemma 1. *For all natural n for any symmetric function $f(x_1 \ldots, x_n)$ the inequality $L(f) \geqslant \min(k(f),\ n - k(f) + 2)$ is true.*

Lemma 2. *For all natural n for any symmetric function $f(x_1 \ldots, x_n)$ the inequality $L(f) \leqslant \min(k(f),\ n - k(f) + 2)$ is true.*

First, in Section 3.1 we shortly describe the idea of the proof of Lemma 1. Then in Sections 3.2 – 3.3 we give a detailed proof. Further, using the ideas from [6], we prove Lemma 2. The proofs of Lemmas 1 and 2 together lead to the proof of Theorem 1.

3.1 Lemma 1: the idea of proof

In this section we explain the idea of the proof of the lower bound from Lemma 1 of a special case. For any symmetric function f consider all circuits computing this function and having the following property: all circuit inputs are connected to inputs of all gates of the circuit. (In Section 3.4 we show that any symmetric function may be computed by a circuit of such type.) For any circuit S_f with the property described we prove that the lower bound on its complexity is $L(S_f) \geqslant k(f)$. This special case is a good example to demonstrate roughly the idea of the proof of Lemma 1. Proving the lemma in Sections 3.2 – 3.3 in general case, we do not suppose circuits to have the above property.

For the special case under study we determine the notion of a first non-zero gate on a given input. On a given tuple $\boldsymbol{\alpha} \in \{0,1\}^n$ let the gates $e_{i_1}, \ldots, e_{i_k}$ be all gates of the circuit S_f such that $h_{i_j}(\boldsymbol{\alpha}) = 1$ for all $j \in [k]$. Then the gate e_m is called a first non-zero gate on the tuple $\boldsymbol{\alpha}$ if it has the minimal index among the gates $e_{i_1}, \ldots, e_{i_k}$. Note that this definition is quite intuitive.

Take any chain $\widetilde{C}$ consisting of $n+1$ different tuples of the cube $\{0,1\}^n$.The chain $\widetilde{C}$ has the following property: no gate of the circuit S_f occurs to be twice a first non-zero gate on two different tuples of this chain. Indeed, suppose the opposite: there is a first non-zero gate e_t on tuples $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ of the chain $\widetilde{C}$. Without loss of generality suppose that the tuple $\boldsymbol{\beta}$ is less than the tuple $\boldsymbol{\gamma}$. By the definition of a first non-zero gate, all gates with the indices less than t output 0 on the tuples $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$. That is, the antichain function g_t which corresponds to the gate e_t outputs 1 on a pair of two comparable tuples. This contradicts the fact that g_t is an antichain function.

From the property proved it follows that the complexity of the circuit S_f is less then or equal to the number of tuples of the chain $\widetilde{C}$ such that first non-zero gates occur on these tuples. Note that if the function f equals 1 on a given tuple, then there exists a first non-zero gate in the circuit S_f on this tuple. This leads to the lower bound $L(S_f) \geqslant k(f)$ on the circuit complexity.

When we prove the above property we use the fact that the circuit S_f is of a special type: all circuit inputs are connected with the inputs of the gate e_t corresponding to the function g_t. In the general case a circuit does not have such a property and the statement similar to above cannot be proved for an arbitrary chain. Furthermore, in general case the notion of a first non-zero gate has to be defined in a more sophisticated manner. These technical peculiarities allow to prove Lemma 1 in the general case basing on the main ideas introduced above.

3.2 General construction.

Proof of Lemma 1. Consider an arbitrary symmetric function $f(x_1, \dots, x_n)$ such that it equals 1 on the tuples from $k(f)$ different layers of the cube $\{0,1\}^n$. Consider an arbitrary circuit S in the AC basis computing the function f. Let $L(S) = s$ and let $e_1, \dots, e_s$ be all the circuit gates. Antichain functions $g_1, \dots, g_s$ are assigned correspondingly to these gates. Let us prove the bound $s \geqslant \min\big(k(f),\ n - k(f) + 2\big)$.

To prove the lemma, we construct a chain C consisting of $n+1$ different tuples of the cube $\{0,1\}^n$. We take the top and the bottom tuples of the cube as boundary tuples of the chain C, i.e. the tuples $\mathbf{1}, \mathbf{0} \in \{0,1\}^n$. In what follows we construct internal tuples of the chain.

Informally, we obtain internal tuples by moving in the cube in two directions: descending from the top tuple or ascending from the bottom one. We descend as follows. In a current tuple of the chain, by a certain algorithm, we choose its component equal to 1 and flip its value. The tuple obtained we add to the chain. From now on the value of an input variable that corresponds to the chosen component we put be equal to 0 and further it will not be changed. Similarly, we ascend from the bottom tuple: we choose a certain component equal to 0 in a current tuple and flip its value. Thus, we obtain a new tuple of the chain C. We put be the corresponding input variable equal to 1. Each time when we fix an input variable by some value we proceed to a subcube of a smaller dimension and further we consider the top and the bottom tuples of this subcube.

Let us continue to describe the construction process in detail. To construct the chain, we consider the gates of the circuit S in the order of a given regular numeration starting from the gate e_1. The following parameters characterize the construction process.

- The number of the step $i \in [s]$; it is determined by a circuit gate which initiates the step.
- Constructing the chain, we simultaneously obtain two sets: $F_i,\ T_i \subseteq [n]$. These are the sets of the indices of the inputs variables whose values after the i-th step we have put equal to 0 (F_i) and 1 (T_i) respectively. We call an input variable of the circuit *free with respect to a set* A, where $A \subseteq [n]$, if its index belongs to the set $[n] \setminus A$. During the chain construction we will consider free variables with respect to sets $F_i \cup T_i$. Informally, these are variables such that up to a certain moment we have not yet define their values to by 0 or 1. We call circuit inputs *free with respect to a set* A, where $A \subseteq [n]$, if the variables assigned to these inputs are free with respect to the set A.
  On the i-th step we consider a subcube with the top tuple $\boldsymbol{x}^{[n]\setminus F_{i-1}}$ and the bottom one $\boldsymbol{x}^{T_{i-1}}$ and we construct a new subcube with the top and bottom tuples $\boldsymbol{x}^{[n]\setminus F_i}$ and $\boldsymbol{x}^{T_i}$ respectively.
- E_i is a set of gates from the set $\{e_1, \dots, e_i\}$ whose inputs may be connected only to outputs of gates with smaller indices and to circuit inputs corresponding to input variables with the indices from the set $F_i \cup T_i$. For any i we have $E_i \subseteq \{e_1, \dots, e_i\}$. On the i-th step we add a circuit gate to the set E_i if none of its inputs are connected to circuit inputs free with respect to the set $F_i \cup T_i$.

Note that the sets F_i, T_i and E_i are not decreasing in i. At the beginning of the construction we put $i = 0$, $F_0 = T_0 = \varnothing$, $E_0 = \varnothing$ and $C = \{\mathbf{0}, \mathbf{1}\}$ as mentioned above.

Now we define the key notion of the proof.

Definition 1. Consider the set of gates $E \subseteq \{e_1, \dots, e_s\}$. Consider an arbitrary input $\boldsymbol{\alpha} \in \{0,1\}^n$ of the circuit S. Let $e_{r_1}, \dots, e_{r_l}$ be all circuit gates such that for any $j \in [l]$

1) $h_{r_j}(\boldsymbol{\alpha}) = 1$;
2) $e_{r_j} \in \{e_1, \dots, e_s\} \setminus E$.

We call the gate e_m *first non-zero gate on the tuple* $\boldsymbol{\alpha}$ *with respect to the set* E if it has the minimal index among the gates $e_{r_1} \dots, e_{r_l}$.

Sometimes when we discuss a first non-zero gate on a given input we will not specify with respect to what set it is considered as far as it will be clear from the context.

We construct the chain in such a way that for any i after the i-th step the following **properties** are fulfilled.

1. For all $t, p \leqslant i$: $F_t \cap T_p = \varnothing$ (i.e. we can not put a variable to be equal to 1 and later change it to 0; it was mentioned earlier informally).

2. For any gate $e_j \in \{e_1, \dots, e_i\} \setminus E_i$ and for a function h_j computed on the output of the gate e_j the following equality takes place:
$$h_j(\boldsymbol{x}^{[n]\setminus F_i}) = h_j(\boldsymbol{x}^{T_i}) = 0.$$

Property 2 plays a crucial role in the construction process. Informally, this property means that if we input the top and the bottom tuples of a current subcube after the i-th step to the circuit inputs then all circuit gates except the gates from the set E_i output 0 on these two tuples.

There are two stages in the construction of the chain.

Stage 1. The stage consists of s steps, by the number of circuit gates. Suppose i steps are done, we describe the $(i+1)$-th step.

At the end of the i-th step the first i circuit gates $e_1, e_2, \dots, e_i$ are considered. We have constructed a certain set F_i which includes the indices of the input variables that we have put be equal to 0. Similarly, we have constructed a set T_i consisting of the indices of the input variables that we have put be equal to 1. The chain C contains the tuples $\mathbf{0}, \mathbf{1} \in \{0,1\}^n$ as well as the tuples that we have obtained simultaneously with constructing F_i and T_i. We have also constructed a set E_i. Now we consider the $(n - |F_i| - |T_i|)$-dimensional subcube with the top and the bottom tuples $\boldsymbol{x}^{[n]\setminus F_i}$ and $\boldsymbol{x}_i^T$ respectively.

Starting the $(i+1)$-th step, we proceed to the gate e_{i+1}. All possible cases are considered below. Note they can not occur simultaneously.

1. The inputs of the gate e_{i+1} are connected directly to none of the free circuit inputs with respect to the set $F_i \cup T_i$. Let $E_{i+1} = E_i \cup \{e_{i+1}\}$, $F_{i+1} = F_i$, $T_{i+1} = T_i$. The $(i+1)$-th step is finished.
2. An input of the gate e_{i+1} is connected directly to a free circuit input with respect to the set $F_i \cup T_i$. Let a variable x_m, where $m \in [n] \setminus (F_i \cup T_i)$, be assigned to this circuit input. Let us verify whether Property 2 is true for the gate e_{i+1}, i.e. the equation $h_{i+1}(\boldsymbol{x}^{[n]\setminus F_i}) = h_{i+1}(\boldsymbol{x}^{T_i}) = 0$ takes place.
   2.1. If Property 2 is true, then we put $E_{i+1} = E_i$, $F_{i+1} = F_i$, $T_{i+1} = T_i$. The $(i+1)$-th step is finished.
   2.2. Assume Property 2 for the gate e_{i+1} is not true for the top tuple of the current subcube (the bottom tuple case will be described later). That is, $h_{i+1}(\boldsymbol{x}^{[n]\setminus F_i}) = 1$. For convenience let us divide this case into two substeps.
      2.2.1. At first we start with the gate e_{i+1}. By the condition of the case 2.2, the gate e_{i+1} is a first non-zero one on the tuple $\boldsymbol{x}^{[n]\setminus F_i}$ with respect to the set E_i. Let the above-mentioned variable x_m be equal to 0.
      We will use some auxiliary notations: F_{i+1}^r, E_{i+1}^r and later T_{i+1}^r, where r is a natural parameter.
      For $r = 1$ let $F_{i+1}^1 = F_i \cup \{m\}$. Let also $T_{i+1} = T_i$. That is, we proceed to a subcube of a smaller dimension $n - |F_{i+1}^1| - |T_{i+1}|$. We add the top tuple $\boldsymbol{x}^{[n]\setminus F_{i+1}^1}$ of this subcube to the chain C. Then we put E_{i+1}^1 be equal to the union of the set E_i and the set of gates e_j, where $j \leqslant i+1$, such that when we put the value of the variable x_m to be equal to some value, their inputs will be no longer connected to any free circuit inputs with respect to the set $F_{i+1}^1 \cup T_{i+1}$.
      2.2.2. Further, we proceed to the gates e_j, where $j \leqslant i+1$. At first, we still consider the value of r to be equal to 1.
      Consider the gates e_j, where $j \leqslant i+1$ and $e_j \notin E_{i+1}^r$. Let us verify whether Property 2 is true for them on the tuple $\boldsymbol{x}^{[n]\setminus F_{i+1}^r}$. Suppose we found gates such that Property 2 does not take place on them. Let e_l be a gate with the minimal index among these gates.

      Lemma 3. *At least one input of the above-mentioned gate e_l is connected directly to a free circuit input with respect to the set $F_{i+1}^r \cup T_{i+1}$.*

      Proof. For the function h_l computed on the output of the gate e_l we have $h_l(\boldsymbol{x}^{[n]\setminus F_{i+1}^r}) = 1$, $h_l(\boldsymbol{x}^{T_{i+1}}) = 0$. The value of the antichain function g_l corresponding to the gate e_l is determined by values on some circuit inputs as well as by values on the gates of two types:

      1) gates e_u such that $e_u \notin E_{i+1}^r$, where $u < l$,

2) gates e_u such that $e_u \in E^r_{i+1}$, where $u < l$.

Values on the outputs of the 2-type gates are determined by values on the outputs of the 1-type gates. Since l is minimal, all 1-type gates e_j output 0 on the top and the bottom tuples of the subcube. Therefore, 2-type gates output the same values on these two tuples. That is, the function g_l outputs two different values on a pair of tuples with the following property: in these tuples the components corresponding to the values of gates with indices smaller than l are equal. Therefore, the values of some other components in these tuples should be different. That is, there exists a free circuit input with respect to the set $F^r_{i+1} \cup T_{i+1}$ and it is connected directly to an input of the gate e_l. Lemma 3 is proved. □

Let us proceed with an argument. By Lemma 3, the gate e_l is a first non-zero one on the tuple $\boldsymbol{x}^{[n]\setminus F^r_{i+1}}$ with respect to the set E^r_{i+1}. Consider any free circuit input with respect to the set $F^r_{i+1} \cup T_{i+1}$ such that it is connected directly with some input of the gate e_l. Put a variable assigned to this input to be equal to 0. That is, we add the index of this variable to the set F^r_{i+1}. The set obtained we denote by F^{r+1}_{i+1}. That is, we proceed to the subcube of a smaller dimension. We add to the chain C the top tuple of this subcube $\boldsymbol{x}^{[n]\setminus F^{r+1}_{i+1}}$.
Let E^{r+1}_{i+1} be equal to the union of the set E^r_{i+1} and the set of all gates e_j, where $j \leqslant i+1$, such that their inputs are no longer connected directly to any circuit inputs free with respect to the set $F^{r+1}_{i+1} \cup T_{i+1}$.

Further, we start a cycle: we perform the process described in 2.2.2 for all $r = 2, \ldots, q$, where q is a value of the parameter r such that Property 2 is true for all gates e_j, where $j \in [i+1]$ and $e_j \notin E^q_{i+1}$. Note an important property that will be used further.

Lemma 4. *Any gate e_p, where $p \in [i+1]$, such that during the process described in 2.2 it was added to the set E^r_{i+1}, where $r \leqslant q$, outputs 0 on all tuples of the $\big(n - |F^r_{i+1}| - |T_{i+1}|\big)$-dimensional subcube on which gates e_a, where $a < p$, $e_a \notin E^r_{i+1}$, output 0.*

Proof. No inputs of the gate e_p are connected to circuit inputs free with respect to the set $F^r_{i+1} \cup T_{i+1}$. That is, this gate outputs the same value on all above-mentioned tuples. Since $T_{i+1} = T_i$, it outputs 0 on the bottom tuple of the subcube. Hence, this gate also outputs 0 on all tuples described. Lemma 4 is proved. □

After all above actions are performed we put $F_{i+1} = F^q_{i+1}$, $E_{i+1} = E^q_{i+1}$. The $(i+1)$-th step is finished. Now we consider the last possible case.

2.3. Let Property 2 for the gate e_{i+1} be not true on the bottom tuple of the subcube: $h_{i+1}(\boldsymbol{x}^{T_i}) = 1$.

Lemma 5. *Cases 2.2 and 2.3 cannot occur simultaneously.*

Proof. Suppose the opposite. For all gates $e_j \notin E_i$, where $j < i+1$, functions h_j output 0 on the tuples $\boldsymbol{x}^{[n]\setminus F_i}$ and $\boldsymbol{x}^{T_i}$. For all gates $e_j \in E_i$ with $j < i+1$ functions h_j output the same on both tuples. The tuples $\boldsymbol{x}^{[n]\setminus F_i}$ and $\boldsymbol{x}^{T_i}$ are comparable. By the conditions of the case 2, there is an input of the gate e_{i+1} which is connected to a circuit input such that it has different values in these two tuples. Hence, the function g_{i+1} outputs 1 on a pair of comparable tuples that differ at least in one component. This contradicts the fact that g_{i+1} is an antichain function. Lemma 5 is proved. □

Now we return to the Case 2.3. It is symmetrical to the Case 2.2. So, we proceed similarly with the only difference that we let free variables be equal to ones instead of zeros. First, put a variable x_k be equal to 1 and $T^1_{i+1} = T_i \cup \{k\}$ (using the notation introduced in the Case 2.2). Put $F_{i+1} = F_i$. That is, we proceed to a subcube of a dimension $n - |F_{i+1}| - |T_{i+1}|$. We add the tuple $\boldsymbol{x}^{T_{i+1}}$ to the chain C. Then, similarly to the Case 2.2, we perform the cycle described in 2.2.2 to construct respectively the sets T^2_{i+1}, T^3_{i+1} and so on, until for some v we obtain that Property 2 is true for all gates e_j, with $j \leqslant i+1$ such that $e_j \notin E^v_{i+1}$. Then we put $T_{i+1} = T^v_{i+1}$, $E_{i+1} = E^v_{i+1}$. The $(i+1)$-th step is finished.

If as a result of Stage 1 the values of all input variables became fixed, then the chain C consisting of $n+1$ tuples will be constructed and we finish the process.

Suppose we have performed s steps described in the Cases 1 and 2 of Stage 1. That is, we have considered all s gates of the circuit S but the chain is still not constructed. After the s-th step we has obtained the sets F_s, T_s and E_s. For simplicity, from now on we omit indices and write F, T and E respectively. Now we proceed to Stage 2.

Stage 2. The original n-dimensional cube is restricted to a subcube of a dimension $n-|F|-|T|$ such that all the circuit gates $e_1,\dots,e_s$, except those from the set E, output 0 on the top tuple $\boldsymbol{x}^{[n]\setminus F}$ and on the bottom one $\boldsymbol{x}^{T}$ of this subcube. Let $T^0 = T$. Then we add to T^0 the indices of variables from the set $[n]\setminus(F\cup T)$. That is, we obtain the sets T^1, T^2 and so on as follows:

1) if $i \geqslant 0$ and there is no first non-zero gate on the tuple $\boldsymbol{x}^{T^i}$ with respect to the set E, then we add to the set T^i the index of any variable from the set $[n]\setminus(F\cup T^i)$. We obtain a set T^{i+1} and a tuple $\boldsymbol{x}^{T^{i+1}}$ which we add to the chain;
2) if $i \geqslant 0$ and there exists a first non-zero gate on the tuple $\boldsymbol{x}^{T^i}$ with respect to the set E, then, similarly to the proof of Lemma 3, it is easy to show that there exists an input of this gate connected to a circuit input free with respect to the set $[n]\setminus(F\cup T^i)$. We put the free input variable corresponding to that circuit input to be equal to 1. Thus we obtain a set T^{i+1} and a new tuple $\boldsymbol{x}^{T^{i+1}}$ which we add to the chain.

If during the above process when we add a variable to the set T^i and obtain a gate e_d such that its inputs are no longer connected to circuit inputs free with respect to the set $F\cup T^i$, then we put $E = E\cup\{e_d\}$. Clearly, all such gates have a property similar to one from Lemma 4. Note that each time we consider a first non-zero gate with respect to a current set E.

We repeat the above process until all free input variables became fixed to some values, i.e. until we construct the chain required. Note that in Stage 2 we choose the set T to fix the idea, we could also choose the set F and put variables to be equal to zeros respectively.

When constructing the chain C we feed to the circuit inputs only tuples of types $\boldsymbol{x}^{T}$ and $\boldsymbol{x}^{[n]\setminus F}$(for brevity, indices are omitted). These tuples are comparable, since on each step $T\subseteq[n]\setminus F$. Thus, the set C is indeed a chain.

3.3 Lemma 1: completion of the proof.

The chain C has the following property.

Lemma 6. *None of the gates of the circuit S occur to be first non-zero (with respect to the corresponding sets) twice on the tuples of the chain C.*

Proof. The proof is by contradiction. Without loss of generality that assume that the tuple $\boldsymbol{x}^{P}$ is added to the chain on the step b and the tuple $\boldsymbol{x}^{P'}$ is added on the step c, where $b<c$. On these steps the sets E_b and E_c are constructed. Moreover, $E_b\subseteq E_c$. Assume there is an index $t\in[s]$ such that the gate e_t is a first non-zero gate on the tuples x^P and $x^{P'}$ with respect to the corresponding sets. In particular, $h_t(\boldsymbol{x}^P) = h_t(\boldsymbol{x}^{P'}) = 1$. For any gate $e_j\notin E_c$, where $j<t$, by the definition of first non-zero gate (with respect to the set E_c and hence, the set E_b) the following equality takes place: $h_j(\boldsymbol{x}^P) = h_j(\boldsymbol{x}^{P'}) = 0$. Also for any gate $e_j\in E_c\setminus E_b$, where $j<t$, by the property from Lemma 4, we obtain: $h_j(\boldsymbol{x}^P) = h_j(\boldsymbol{x}^{P'}) = 0$. Values of gates $e_j\in E_b$, where $j<t$, are determined by values of other gates with smaller indices. Therefore, values of any gates with the indices less than t are equal on the tuples $\boldsymbol{x}^{P'}$ and $\boldsymbol{x}^{P}$. By the construction, the tuple $\boldsymbol{x}^{P'}$ differs from the tuple $\boldsymbol{x}^{P}$ by the value of at least one component corresponding to an input variable such that it is connected directly with an input of the gate e_t.

That is, the antichain function g_t corresponding to the gate e_t outputs 1 on a pair of comparable tuples. This contradicts the fact that g_t is an antichain function. Lemma 6 is proved. □

Let us finish the proof of Lemma 1. Recall that the constructed chain C contains $n+1$ tuplesssssssssss of the cube $\{0,1\}^n$. The circuit outputs the same value on all tuples of C such that there are no first non-zero gates on them with respect to a corresponding set. Denote this value by $a \in \{0,1\}$. There exists a first non-zero gate on any tuple of the chain on which the circuit outputs $1-a$. The function $f(x_1,\dots,x_n)$ takes value 1 on $k(f)$ tuples of this chain and value 0 on $n+1-k(f)$ tuples. There are two possible cases.

1. If $a=0$, then for any tuple on which the circuit outputs 1 there exists a first non-zero gate in the circuit. That is, there are first non-zero gates on $k(f)$ tuples of the chain C. By Lemma 6, all these gates are different. Therefore, $L(S) \geqslant k(f)$.
2. If $a=1$, then for any tuple on which the circuit outputs 0 there exists a first non-zero gate in the circuit. That is, there are first non-zero gates on $n+1-k(f)$ tuples of the chain C. Similarly, by Lemma 6, we obtain: $L(S) \geqslant n+1-k(f)$. Moreover, the last circuit gate e_s is not the first non-zero one, i.e. if there is a first non-zero gate on a tuple, then the circuit outputs 0 on it, and the gate e_s as well. Therefore, $L(S) \geqslant n+2-k(f)$.

We have shown that $L(S) \geqslant \min\big(k(f),\, n-k(f)+2\big)$. Since the circuit S is arbitrary, we obtain the similar inequality for $L(f)$. Lemma 1 is completely proved.

3.4 Proof of Lemma 2.

Consider an arbitrary symmetric Boolean function $f(x_1,\dots,x_n)$. We split the proof into two parts. In the first one we will prove the bound $L(f) \leqslant k(f)$ and then we will show the bound $L(f) \leqslant n-k(f)+2$. These two bounds together will give us the statement required.

Proof. 1. Let the function f be equal to 1 on layers with indices $i_1,\dots,i_{k(f)}$, where $i_1 < \dots < i_{k(f)}$.
Recall that by $\boldsymbol{x}$ we denote a set of values of variables $(x_1,\dots,x_n)$. For a function f, for any $t \in \{i_1,\dots,i_{k(f)}\}$ we define a function $h_t(\boldsymbol{x})$ as follows: $h_t(\boldsymbol{x})=1$ iff $\sum_{p=1}^n x_p = i_t$. Clearly, all the functions h_t are antichains since they are characteristic functions of layers of the cube.
Denote by $\boldsymbol{y}$ a set of values of variables $(y_1,\dots,y_{k(f)-1})$.
Let M_1 be a set of tuples $(\boldsymbol{y},\boldsymbol{x})$ such that three properties take place simultaneously: 1) there exists $j \in [k(f)-1]$ such that $y_j=1$; 2) $y_q=0$ for all $q \neq j$; 3) $\sum\limits_{p=1}^{n} x_p = i_j$. Let M_2 be a set of tuples $(\boldsymbol{y},\boldsymbol{x})$ such that the following properties take place: $y_q=0$ for all $q \in [k(f)-1]$ and $\sum_{p=1}^n x_p = i_{k(f)}$.
Define a function g of $k(f)-1+n$ variables $y_1\dots,y_{k(f)-1},x_1\dots,x_n$ as follows: $g(\boldsymbol{y},\boldsymbol{x})=1$ iff the tuples $(\boldsymbol{y},\boldsymbol{x})$ belong to the set $M_1 \sqcup M_2$.
It is easy to see that the function g is antichain. Indeed, consider two tuples from the support of g : $(\boldsymbol{y}_1,\boldsymbol{x}_1) \neq (\boldsymbol{y}_2,\boldsymbol{x}_2)$. If $(\boldsymbol{y}_1,\boldsymbol{x}_1),(\boldsymbol{y}_2,\boldsymbol{x}_2) \in M_1$ and $\boldsymbol{y}_1 \neq \boldsymbol{y}_2$, then the tuples are incomparable. If $(\boldsymbol{y}_1,\boldsymbol{x}_1),(\boldsymbol{y}_2,\boldsymbol{x}_2) \in M_1$ and $\boldsymbol{y}_1 = \boldsymbol{y}_2$, then the indices of the components where i_j ones appear in the tuple $\boldsymbol{x}_1$ differ from those in the tuple $\boldsymbol{x}_2$. That is, the tuples $(\boldsymbol{y}_1,\boldsymbol{x}_1)$ and $(\boldsymbol{y}_2,\boldsymbol{x}_2)$ are incomparable. If $(\boldsymbol{y}_1,\boldsymbol{x}_1),(\boldsymbol{y}_2,\boldsymbol{x}_2) \in M_2$, then, similarly to the previous case, it is easy to see that the tuples are incomparable. If without loss of generality $(\boldsymbol{y}_1,\boldsymbol{x}_1) \in M_1$, $(\boldsymbol{y}_2,\boldsymbol{x}_2) \in M_2$, then $\boldsymbol{y}_1 > \boldsymbol{y}_2$ and there are more ones in $\boldsymbol{x}_2$ than in $\boldsymbol{x}_1$. Therefore, the tuples $(\boldsymbol{y}_1,\boldsymbol{x}_1)$, $(\boldsymbol{y}_2,\boldsymbol{x}_2)$ are incomparable.
Let us compute the function $f(\boldsymbol{x})$ as follows:

$$f(\boldsymbol{x}) = g\left(h_1(\boldsymbol{x}),\dots,h_{k(f)-1}(\boldsymbol{x}),\boldsymbol{x}\right). \tag{1}$$

That is, if we feed to the function g the characteristic functions of the layers whose union forms the support of f, except the function $h_{k(f)}$, then we obtain a realization of the function f.
Let us verify the equality (1). Consider an arbitrary tuple $\boldsymbol{\alpha}=(\alpha_1,\dots,\alpha_n)$. There are two possibilities.
A. $f(\boldsymbol{\alpha})=1$, then $\sum_{p=1}^n \alpha_p \in \{i_1,\dots,i_{k(f)}\}$. If $\sum_{p=1}^n \alpha_p \leqslant i_{k(f)-1}$, then there exists an index $j \in [k(f)-1]$ such that $h_j(\boldsymbol{\alpha})=1$ and $h_q(\boldsymbol{\alpha})=0$, and $\sum_{p=1}^n \alpha_p = i_j$ for all $q \neq j$. Thus, the tuple $(\boldsymbol{y},\boldsymbol{\alpha})$,

where $(y_1, \dots, y_{k(f)-1}, \boldsymbol{\alpha}) = \left(h_1(\boldsymbol{\alpha}), \dots, h_{k(f)-1}(\boldsymbol{\alpha}), \boldsymbol{\alpha}\right)$, belongs to the set M_1. Hence, $g(\boldsymbol{y}, \boldsymbol{\alpha}) = 1$. If $\sum_{p=1}^{n} \alpha_p = i_{k(f)}$, then $y_q = h_q(\boldsymbol{\alpha}) = 0$ for all $q \in [k(f) - 1]$. That is, the tuple $(\boldsymbol{y}, \boldsymbol{\alpha})$ belongs to the set M_2 and therefore $g(\boldsymbol{y}, \boldsymbol{\alpha}) = 1$.

B. $f(\boldsymbol{\alpha}) = 0$, then $h_j(\boldsymbol{\alpha}) = 0$ for all $j \in [k(f)]$. In particular, $h_{k(f)}(\boldsymbol{\alpha}) = 0$. Thus, $\sum_{p=1}^{n} \alpha_p \neq i_{k(f)}$. It follows that the tuple $(\boldsymbol{y}, \boldsymbol{\alpha}) = \left(h_1(\boldsymbol{\alpha}), \dots, h_{k(f)-1}(\boldsymbol{\alpha}), \boldsymbol{\alpha}\right)$ belongs neither to M_1, nor to M_2. Therefore, $g(\boldsymbol{y}, \boldsymbol{\alpha}) = 0$.

From the equality (1) it follows that the function f may be computed in the AC basis by a circuit of the complexity at most $k(f)$.

That is, for any symmetric function f there exists a circuit S of the complexity $L(S_f) \leqslant k(f)$ such that it computes f and has the following property: for any gate of the circuit its inputs are connected to all the circuit inputs.

2. From the proof of the Case 1 it follows that for the function $\overline{f}$ the inequality $L(\overline{f}) \leqslant k(\overline{f})$, where $k(\overline{f}) = n + 1 - k(f)$, takes place. That is, the function $\overline{f}$ may be computed by a circuit of the complexity at most $n + 1 - k(f)$. By a gate computing negation, from any circuit computing $\overline{f}$ we can easily obtain a circuit computing f. That is, $L(f) \leqslant n - k(f) + 2$.

Hence, for an arbitrary symmetric function f, we have shown that $L(f) \leqslant \min(k(f), n - k(f) + 2)$. So, Lemma 2 is proved. □

The proofs of Lemmas 1 and 2 together give the proof of Theorem 1.

4 Proof of Theorem 2

For the majority function $k(m_n) = \left\lfloor \frac{n}{2} \right\rfloor + 1$. By Theorem 1, we obtain: $L(m_n) = \left\lfloor \frac{n}{2} \right\rfloor + 1$. For the parity function $k(p_n) = \left\lfloor \frac{n+1}{2} \right\rfloor$. Correspondingly, by Theorem 1, we obtain: $L(p_n) = \left\lfloor \frac{n+1}{2} \right\rfloor$. This proves Theorem 2.

Note that the maximum of the upper bound from Lemma 2 is achievable: for odd n it is equal to the complexities of the functions m_n, p_n and $\overline{p}_n$, and for even n, it is equal to the complexities of the functions m_n and $\overline{p}_n$.

Acknowledgment: The author is grateful to professor O. M. Kasim-Zade for stating the problem and constant attention. The author is also grateful to A. V. Kochergin for helpful comments.

References

[1] Kasim-Zade O. M., "On the complexity of circuits in an infinite basis", *Vestnik Moskovskogo Un-ta. Ser. 1. Matem. Mekh.*, **6** (1994), 40-44 (in Russian).

[2] Kasim-Zade O. M., "On the complexity of the realization of Boolean functions by circuits in an infinite basis", *Diskret. Anal. Issled. Oper.*, **2**:1 (1995), 7-20 (in Russian).

[3] Lupanov O. B., "A method of circuit synthesis", *Izvestiya Vysshikh Uchebnykh Zavedenii. Radiofizika*, **1**:1 (1958), 120-140 (in Russian).

[4] Lupanov O. B., *Asymptotic Estimates of Complexity of Control Systems*, Izd-vo Moskovskogo Un-ta, 1984 (in Russian).

[5] Podolskaya O. V., "On the circuit complexity lower bounds in antichain basis.", *Vestnik Moskovskogo Universiteta. Seriya 1. Matematika. Mekhanika*, **68**:2 (2013), 17-23 (in Russian).

[6] Podolskaya O. V., "On the complexity of circuits in a certain infinite basis.", Proc. 9th Young Scientists School on Discrete Math. and Appl. (Moscow, 2013), 2013, 97-100 (in Russian).