A set of disjunctions of some variables is constructed and a nonlinear lower bound is proved for the circuit complexity of this set in systems of functional elements (s.f.e.)* in a fixed monotone basis. The proposed method for proving the lower bound of circuit complexity in the s.f.e. differs from previously known methods (in a monotone basis).

1. A monotone Boolean function is the composition of conjunctions and disjunctions. In [1-4] the authors examine the problem of finding a lower bound for the computational complexity of a set of monotone Boolean functions in systems of functional elements (s.f.e.) in a fixed monotone basis, i.e., a basis of monotone functions. Nechiporuk [1] was the first to construct a set of disjunctions with a nonlinear lower bound for their computational complexity in s.f.e. in a monotone basis. The disjunctions constructed by Nechiporuk had the property that no two had more than one common variable. The set of disjunctions in [2] has the same property.

In the present paper we construct a set of disjunctions for which we prove a nonlinear strict (to within a multiplicative constant) lower bound for the computational complexity in s.f.e. in a fixed monotone basis. The method of proving the lower bound may therefore also be of interest. This method was used to obtain the lower bound for the computational complexity of a set of linear forms (see [5, Sec. 1]).

Nechiporuk (see [1]) proved that in computing a set of disjunctions in s.f.e. in a monotone basis we can, without increasing the complexity, restrict ourselves to a basis consisting of one two-place disjunctive. We represent each s.f.e. in the usual way (see [5]) in the form of a directed graph with the number of vertices equal to the number of elements of the s.f.e., counting the variables which occur in it. Each vertex of the graph therefore corresponds to a Boolean function. We shall say that the functions corresponding to the vertices of a graph are computed by a given s.f.e. The complexity of an s.f.e. is the number of vertices in its corresponding graph. Therefore, we shall consider systems of functional elements in a basis of one two-place disjunction, which compute sets of disjunctions.

2. We proceed to the construction of the required set of disjunctions. Let \( M \) be a natural number, \( 1 \) be any nonempty subset of the set \( \{1, \ldots, M\} \). We denote by \( A_1 \) the disjunction of all variables \( x_i (0 \leq i < 2^M) \), such that if \( i = i_1 \ldots i_M \) is the expansion of \( i \) in the binary scale, then the sum \( \sum_{i=1}^{i_M} i_j \) is even. Thus, we construct the set \( \{A_1\}_{\varphi \in \{1, \ldots, M\}} \), consisting of \( 2^M - 1 \) disjunctions of \( 2^M \) variables. We shall calculate the computational complexity of this set in s.f.e. in the basis \( \{V\} \).

The number of variables that occur in any disjunction \( D \) is the weight of \( D \), and we denote it by \( |D| \). We note that for every \( I \neq \emptyset, I \subseteq \{1, \ldots, M\}, |A_I| = 2^{M-1} \) holds.

First we obtain an upper bound for the computational complexity of \( \{A_I\}_{\varphi \in \{1, \ldots, M\}} \) in an s.f.e. in the basis \( \{V\} \). We denote by \( K^\varepsilon_I (\varepsilon = 0, 1) \) the following subset of the set \( \{0, 1, \ldots, 2^M - 1\} \): \( K^\varepsilon_I = \{i; i = i_1 \cdots i_M \text{ is the binary expansion of } I \text{ and } \sum_{i=1}^{i_M} i_j \equiv \varepsilon (\text{mod } 2)\} \).

We put \( A^{\varepsilon}_I = \bigvee_{i \in K^\varepsilon_I} x_i \), where \( \varepsilon \neq 0, 1 \). Obviously, \( A^\phi_I = A_I \) for every \( \varphi \in \{1, \ldots, M\} \). We construct in the basis \( \{V\} \) the s.f.e. of complexity \( \leq M \cdot 2^{M-1} \) which computes the set of disjunctions \( \{A^{\varepsilon}_I\}_{\varphi \in \{1, \ldots, M\}} \), \( \varepsilon = 0, 1 \). We produce the construction by induction on \( M \). For \( M = 1 \) the construction is obvious. Let the required s.f.e. be already constructed for \( M = K \).

*Translator's note: For "functional elements" read "Boolean circuits."
We take two samples of $S$ (denoted by $S_1$ and $S_2$), and we replace every index $i$ of the variable occurring in $S_1$ by $2i$. Correspondingly, in $S_2$ we replace $i$ by $2i + 1$. We denote the systems thus formed by $S'_1$ and $S'_2$, respectively. We denote the output of the s.f.e. $S'_1$ by $\{A^0_1\}$, and the output of $S'_2$ by $\{A^1_1\}$. We now combine $S'_1$ and $S'_2$ and obtain an s.f.e. with input variables $x_0, x_1, \ldots, x_{2k+1}$. We construct the output for the resulting s.f.e., $\{A^i_1\} \in \{1, \ldots, k+1\}$, $i \in 0, \ldots, k$. (see clarification below)

(where $A^0_0$ is the disjunction of all the variables, and $A^1_0$ is the empty disjunction). Let $I \in \{1, \ldots, k+1\}$. We consider two cases:

1) $k+1 \notin I$. Then $A^i_1 = A^0_1 \lor A^1_1$ and $A^i_1 = A^0_1 \lor A^1_1$.

2) $k+1 \in I$. We put $I' = I \setminus \{k+1\}$. Then

$$A^0_1 = A^0_1 \lor A^1_1 \text{ and } A^i_1 = A^0_1 \lor A^1_1.$$

Thus, we have constructed an s.f.e. (see Fig. 1) on the basis $\{V\}$ with complexity no greater than $2^k + k \cdot 2^k = (k+1)2^{k+2}$, which computes the set $\{A_1\} \in \{1, \ldots, k+1\}$, $i \in 0, \ldots, k$. This proves the inductive statement for $M = k+1$.

3) We proceed to establish the lower bound.

**THEOREM.** The computational complexity of the set $\{A_1\} \in \{1, \ldots, k+1\}$ in s.f.e. in a monotone basis is not less than $(M-1)(2^M-1)/2$. As a preliminary, we prove some subsidiary lemmas.

A directed graph $G$ is ordered if it has the following properties:

1) It contains no directed cycles (see [6]);
2) no more than two arcs enter each vertex of the graph $G$.

The letter $G$ will denote ordered graphs.

Vertices that are entered by no arc are input; vertices from which no arc leaves are output. We shall say that the vertex $B$ is situated above the vertex $C$ in the graph $G$, if there exists a directed chain in the graph $G$ from $B$ to $C$ (see [6]). For every vertex $A$ of the graph $G$ we put $\rho_G(A)$ equal to the number of input vertices of the graph $G$ which are situated above $A$ in the subgraph $G'$ of the graph $G$. The following lemma is well-known in coding theory.

**LEMMA 1.** Let $G'$ be a subtree of an ordered graph $G$, having one outgoing vertex $R$. Then

$$\sum_{A \in G} \beta_{G'}(A) \geq \beta_{G'}(R) \log_2 \beta_{G'}(R).$$

Lemma 1 can be proved by induction on the number of vertices of $G'$, using the convexity of the function $x \log_2 x$ on the positive semiaxis.

**LEMMA 2.** Let $G$ be an ordered graph with a single output vertex $R$. Then

$$\sum_{A \in G} \beta_G(A) \geq \beta_G(R) \log_2 \beta_G(R).$$

**Proof.** Let $G = G_0, G_1, \ldots, G_m$ be some chain of ordered graphs, such that

1) $G_i$ is obtained from $G_{i-1}$ by the deletion of some arc $i = 1, \ldots, m$;
2) $G_m$ has a unique output vertex $R$ and is a tree.

It is not difficult to construct such a chain from $G$. For every vertex $A$ of the graph $G$, the inequality

$$\beta_G(A) \geq \beta_G(\hat{A}) \geq \cdots \geq \beta_{G_m}(\hat{A})$$

is satisfied. In addition, by virtue of property (2), for $G_m$ the equality $\beta_G(R) = \beta_{G_m}(R)$ is satisfied. Taking account of this, and applying Lemma 1 for the graph $G_m$, we get