Some Bounds on the Complexity of Predicate Recognition by Finite Automata

Y. BREITBART*

Department of Computer Science, Technion–Israel Institute of Technology, Haifa, Israel

Received May 21, 1974; revised June 13, 1975

Let \( \phi_P(n) \) be the smallest number such that some finite automaton with \( \phi_P(n) \) internal states exists which recognizes predicate \( P \) over the set of words of length not greater than \( n \). Then there exists a predicate \( P \) defined on \((0, 1)\) such that an infinite sequence \( n_1, n_2, \ldots, n_k, \ldots \) when \( n_k \rightarrow \infty \) as \( k \rightarrow \infty \) can be constructed for which \( \phi_T(n_k) \sim 2^{2k+2}/n_k \), where \( T(x) = P(x) \) or \( P(x^R) \), for \( x^R \) is the reverse of \( x \).

1. Introduction

In designing an algorithm to recognize a given predicate, the most basic question is, “What is the inherent computational complexity of the given predicate?”

Unfortunately, determining the inherent computational complexity of a predicate is usually a very difficult task. However, for some classes of algorithms it is possible to construct some complicated predicates. For instance, for a class of finite automata, examples of most complicated predicates were constructed by Trachtenbrot [1] and Grinberg [4].

In the present paper, we also construct a most complicated predicate for recognition by finite automata, which may choose the direction of motion along the input. Before undertaking a precise formulation of the problem, we give a number of definitions.

Let \( A \) be a finite nonempty alphabet. Let \( x = x_1 \cdots x_n \) \((x_i \in A)\) be a word on \( A \). We shall say that the word \( x^R = x_n \cdots x_1 \) is the reverse of \( x \). The set of all words (including empty word \( \lambda \)) on \( A \) will be denoted by \( A^* \). Let us denote the set of words on \( A \) of length less than or equal to \( n \) (equal to \( n \)) by \( A^{\leq n} \) \((A^n)\), respectively. A predicate \( P \) on \( A^* \) is a function mapping \( A^* \) into \( \{0, 1\} \).

A finite automaton over \( A \) is a quadruple \( W = \langle Q, \delta, q_0, F \rangle \), where \( Q \) is a finite set (the set of internal states); \( \delta: Q \times A \rightarrow Q; q_0 \in Q \) (initial state of \( W \)); \( F \subseteq Q \) (the set of
designated final states). The function $\delta$ can be extended to a function $\delta: Q \times A^*$, as follows:

$$\delta(q, \lambda) = q; \text{ where } \lambda \text{ is empty word on } A.$$  

$$\delta(q, xa) = \delta(\delta(q, x), a), \text{ where } q \in Q, a \in A, x \in A^*.$$  

We shall say that a predicate $P(x)$ is recognized by the finite automaton $W = (Q, \delta, q_0, F)$ if the following holds:

$$\forall x (x \in A^* \Rightarrow (P(x) = 1 \Leftrightarrow \delta(q_0, x) \in F))$$

and $n$-recognized if the following is true

$$\forall x (x \in A^n \Rightarrow (P(x) = 1 \Leftrightarrow \delta(q_0, x) \in F)).$$

Let $\phi_p(n)$ be the smallest number such that some finite automaton with $\phi_p(n)$ internal states exists which $n$-recognizes a predicate $P(x)$. 

The function $\phi_p(n)$ was first studied in a somewhat different form by Trachtenbrot [1], who proved that for any $n, \phi_p(n) \leq 2^{n+2}/n$ and for infinitely many $n, \phi_p(n) \leq 2^{n+1}/n$. In the same work, Trachtenbrot constructed examples of predicates such that the above-mentioned bounds may be achieved for an infinite subsequence of $n$. 

Grinberg [4] has established a stronger result, namely, for any $n$ and for any predicate $P, \phi_p(n) \leq c(n) \cdot 2^n/n$, where $2 \leq c(n) \leq 4$ and $\limsup c(n) = 4$ while $\liminf c(n) = 2$. He showed that for any $n$, it is possible effectively to construct a predicate $P(x)$ such that $\phi_p(n) \geq c(n) \cdot 2^n/n$. 

The examples constructed by Trachtenbrot and Grinberg are based on the following simple idea. 

Let $W$ be a finite automaton and $x$ be a word in $A^*$. Then, the information on the value of the predicate for the word $x$ is obtained by the automaton only after reading and recalling almost the whole word. Thus, the automaton must have a sufficiently large number of internal states for remembering “long” words in order to be capable of $n$-recognition of $P(x)$. 

Let us consider the following simple example. Let $A = (0, 1)$ be the alphabet and $P(x)$ be a predicate in alphabet $A$ which is defined as follows. Let $x = a_1 \cdots a_{n-\lceil \log n \rceil + 1} \cdots a_n$ be a word in the alphabet $A$, where $n$ is the length of $x$. Let $z$ denote the suffix of $x$, with length equal to $\lceil \log n \rceil$. Let $|z|$ be a number such that a word $z$ is its binary representation. $P(x)$ is true if, and only if, $a_{|z|+1}$ is equal to 1; for example, $P(0100) = 0$ and $P(1000) = 1$ and so forth. 

If we try to recognize this predicate by finite automata for words of length not greater than $n \geq 2$ ($n$-recognition of the predicate), then we can ensure that any finite automaton that recognizes a predicate $P(x)$ on words of length not greater than $n \geq 2$ must have at least $2^n/n$ internal states. 

Thus, let $W$ be a finite automaton recognizing a predicate $P(x)$ on words of length not greater than $n$. Let $k$ be a number of internal states of $W$. 
Suppose that \( k < \frac{2^n}{n} \). Then, two words \( x = x_1 \cdots x_{n-[\log n]} \) and \( y = y_1 \cdots y_{n-[\log n]} \) exist where \( x \neq y \), such that after reading both of them, \( W \) goes to the same internal state. Since \( x \neq y \), there is a number \( l \) such that \( x_l \neq y_l \). Let \( z_1 \cdots z_{[\log n]} \) be a binary representation of \((l-1)\); then \( P(xz) \neq P(yz) \) but \( T \) for any word \( z \), on words \( xz \) and \( yz \), gives us the same answer. This is impossible. Consequently, \( k \geq \frac{2^n}{n} \).

The examples of predicates constructed on the basis of this idea are such that \( P(x^2) \) are sufficiently simple for their recognition by finite automata; that is, these examples in essence require only that the automata read the word from left to right and therefore, if we place the "basic information" on the right-hand ends of words, then the automata must remember rather long words.

In this work, we allow finite automata to read words either from left to right or from right to left and construct under these conditions an example of maximally "complicated" predicate. In other words, we define a function \( \min(\phi_{P(x)}(n), \phi_{P(x^R)}(n)) \) and prove that a predicate \( P(x) \) and an infinite sequence of \( n \to \infty \) exist, such that

\[
\min(\phi_{P(x)}(n), \phi_{P(x^R)}(n)) \geq 2^{n+2}/n
\]

(generally speaking, we prove a stronger result, namely, that a predicate \( P(x) \) and an infinite sequence of \( n \to \infty \) exist, such that

\[
\phi_{P(x)}(n) = \phi_{P(x^R)}(n) \geq 2^{n+2}/n.
\]

It follows from the results of Trachtenbrot and Grinberg that our result cannot be improved in the following sense: A predicate \( P(x) \) exists, such that

\[
\phi_{P(x)}(n) = \phi_{P(x^R)}(n) \geq 2^{n+2}/n
\]

for almost all \( n \).

The results obtained in this work are analogous in a certain sense to the results of Lupanov and Shannon on the obtaining of asymptotic lower bounds for the realization of a Boolean function by logical networks. At present, in order to obtain lower bounds for the network complexity of Boolean functions of order \( 2^n/n \), there is only one method of counting arguments. However, lower bounds obtained by this method usually are not effective, in the sense that we cannot effectively construct a sequence of functions \( f_n \) such that the complexity of \( f_n \) is more than \( 2^n/n \), although we know that almost all functions require such complexity.

If we use counting arguments in our case, we may obtain analogous estimates; namely, the following observation holds.

**Observation 1.** For any \( n \), a predicate \( P(x) \) exists, such that \( \phi_P(n) \geq 2^{n+1}/n \) states are required for its \( n \)-recognition. Furthermore, for any permutation \( \pi \) (and not only for the reverse of the word), the complexity of \( P(\pi(x)) \geq 2^{n+1}/n \). However, the method of proof (1) does not allow us to obtain an estimate \( 2^{n+2}/n \), and (2) does not give a concrete example of a predicate for which the above-mentioned lower bound is achieved.
As already mentioned, for the case of logical networks, no concrete examples are known of a Boolean function requiring a nonlinear number of elements for its realization. In contrast, for finite automata reading from left to right or from right to left, we may construct examples of predicates with maximal complexity.

2. PRELIMINARY RESULTS

Let \( P(x) \) be a predicate on \( A^* \) and \( n \geq 0 \). We introduce the following relationship on \( A^{<n} \):

\[
x \equiv_P (n) y \iff \forall z (l(xz) \leq n \& l(yz) \leq n \Rightarrow P(xz) = P(yz)),
\]

where \( x, y, z \in A^{<n} \).

If \( x \equiv_P (n) y \), then \( x \) and \( y \) are called \( n \)-equivalent relative to \( P \).

We denote \( \psi_P(n) \) as the greatest number of words from \( A^{<n} \) such that for any pair of words \( x \) and \( y \), \( x \equiv_P (n) y \).

We note the following

**LEMMA 1.** \( \phi_P(n) = \psi_P(n) \).

**Proof.** We prove \( \phi_P(n) \geq \psi_P(n) \), leaving the proof of \( \phi_P(n) \leq \psi_P(n) \) for the reader.

Let \( P(x) \) be a predicate on \( A^* \) and \( n \geq 0 \). Assume that \( P(x) \) is \( n \)-recognized by \( W = (Q, \delta, q_0, F) \) with \( \phi_P(n) \) internal states. Suppose that \( \phi_P(n) < \phi_P(n) \). Then \( x \) and \( y \) in \( A^{<n} \) exist such that \( x \equiv_P (n) y \), but \( \delta(q_0, x) \neq \delta(q_0, y) \). Thus, for any \( z \in A^* \), \( \delta(q_0, xz) = \delta(q, yz) \), i.e., \( P(xz) = P(yz) \), (a) if \( x \equiv_P (n) y \), then \( z \in A^{<n} \) exists such that \( l(xz) \leq n \) and \( l(yz) \leq n \). But \( P(xz) \neq P(yz) \), which contradicts (a).

Therefore, \( \phi_P(n) \geq \psi(n) \).

Henceforth, we always assume that \( A = \{0, 1\} \).

Now, we construct some special Boolean functions which are necessary for the construction of the predicate \( P \), which is complicated in two directions for recognition by finite automata.

Let \( n = 2^k + k \), where \( k = 1, 2, \ldots \).

**LEMMA 2.** For any \( k \), there exists the Boolean function \( F_k \) of \( n \) variables such that for any Boolean function \( f \) of \( k \) variables, there exists a unique pair of words \( x \) and \( y \) (\( l(x) = l(y) = 2^k \)) such that

\[
F_k(x, z) = f(z), \quad F_k(z, y) = f(z).
\]

Proof of this lemma is given in the Appendix. We now consider the following Boolean function of \( n = 2^k + 2k \) variables.

\[
\Phi_A(x, y, z) = \sum_{\alpha, \beta \in \{0, 1\}^k} x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdot \cdots \cdot x_k^{\alpha_k} \cdot y_1^{\beta_1} \cdot \cdots \cdot y_{|\beta|}^{\beta_{|\beta|}} \cdot z_1^{\beta_{|\beta| + 1}} \cdot \cdots \cdot z_k^{\beta_{|\beta| + k}},
\]
where\( x_1, y_1, \alpha, \beta \in (0, 1), x = x_1 \cdots x_k; y = y_1 \cdots y_k ; z = z_1 \cdots z_k; \alpha = \alpha_1 \cdots \alpha_k; \beta = \beta_1 \cdots \beta_k (|\alpha| (|\beta|) \text{ is an integer with binary representation } \alpha(\beta) \text{ and } x^0 = \bar{x} \text{ but } x^1 = x) \).

**Example 1.** Let \( k = 1 \); then \( n = 4 \),

\[
\Phi(x_1, y_1, y_2, z_1) = \bar{x}_1 y_1 \bar{z}_1 + \bar{x}_1 y_2 z_1 + x_1 y_2 \bar{z}_1 + x_1 y_1 z_1.
\]

The following lemma is obvious.

**Lemma 3.** For any \( u \in (0, 1)^k \) and for any Boolean function \( f \) of \( k \) variables, there exists a unique pair of words \( v \) and \( w \) (\( l(v) = l(w) = 2^k \)), such that

\[
\Phi_k(u, v, z) = f(z), \quad \Phi_k(x, w, u) = f(z).
\]

**3. Main Result**

Now, we are ready to prove the following

**Theorem.** There exists a predicate \( P \) defined on \((0, 1)^*\) such that an infinite sequence \( n_1, \ldots, n_k, \ldots \), where \( n_k \to \infty \) as \( k \to \infty \), can be constructed for which

\[
\phi_T(n_k) \sim 2^{n_k+2}/n_k,^2
\]

where

\[
T(x) = P(x) \text{ or } P(x^R).
\]

Before presenting the proof of our theorem, we outline its structure. To prove (1), we evaluate the number \( \psi_T(n) \) by employing the relationship \( \phi_T(n) \geq \psi_T(n) \).

For the proof of our result we build a predicate \( P(x) \) using the Boolean function constructed in the previous section and having the following properties. Let \( n = 2^{2k+1} + 2k - 1 \).

1. Any two words \( x \neq y \), which belong to the set \((0, 1)^{n_2k} \) are not \( n \)-equivalent relative to \( P(x) \) and \( P(x^R) \). This results from Lemma 4.

2. All words, with length between \( n - 4k + 1 \) and \( n - 2k - 1 \) denoted by the set \( S \) are also pairwise not \( n \)-equivalent relative to \( P(x) \) and \( P(x^R) \). This results from Lemma 5.

3. There exists a "very small" number of words in \((0, 1)^{n_2k} \), which are \( n \)-equivalent to words from the set \( S \) relative to \( P(x) \) or \( P(x^R) \).

\( \sum \) sum of modulo 2; \( \oplus \), sum of modulo \( 2^k \).

\( ^2 \) We say that \( \phi(n) \sim \psi(n) \) iff \( \lim \phi(n)/\psi(n) = 1 \) as \( n \to \infty \); we say that \( \phi(n) \preceq \psi(n) \) iff \( \lim \phi(n)/\psi(n) \leq 1 \) as \( n \to \infty \).
COMPLEXITY OF PREDICATES

Therefore, a set of non $n$-equivalent words relative to $P(x)$ and $P(x^R)$ may contain all of the words from $S$ and "almost" all of the words from $(0, 1)^{n-2k}$. This will establish the theorem.

Proof. Let $n_k = 2^{2k+1} + 2k - 1$. The predicate, the existence of which is stated above, will be defined by induction on the length of word

$$P(\lambda) = P(0) = P(1) = 0.$$ 

Let $P(x)$ be defined for all words $x$, such that $l(x) \leq n_{k-1}$.

If $n_{k-1} + 1 \leq l(x) \leq n_k - 2k - 1$, then $P(x) = 0$. (3.1)

Now, in order to complete the definition $P(x)$, we must define $P(x)$ for words $x$ of length $n_k = n, n-1, \ldots, n-2k$, correspondingly.

For $x \in (0, 1)^n$, let $x = usrtz$, where

$$l(u) = l(z) = 2k,$$

$$l(s) = l(t) = 2^{2k-1} - k.$$ (3.2)

We define

$$P(x) = 1 \iff F_{2k}(u, s, t, z) = 1,$$ (3.3)

where $F_{2k}$ satisfies Lemma 2.

For $x \in (0, 1)^{n-j}$, where $j = 1, \ldots, 2k - 1$, let

$$x = usrtwz,$$ (3.4)

where

$$l(u) = l(z) = 2k - j,$$

$$l(v) = l(w) = 2^{2k} - 2^{2k-j} - k + \lfloor j/2 \rfloor + 1,$$ (3.5)

$$l(s) = l(t) = 2^{2k-j-1} - ((1 + (-1)^j)/2).$$

For odd $j$, we define

$$P(x) = 1 \iff \Phi_{2k-j}(u, s, t, z) = 1,$$ (3.6)

where $\Phi_{2k-j}$ satisfies Lemma 3. If $j$ is even, then let

$$v = v^aav^b; \quad w = cw^aaw^t,$$ (3.7)

where letters $(a, d)$ and $(b, c)$ are located symmetrically in word $x$ and

$$l(v^t) = l(w^t) = 2^{2k} - 2^{2k-j+1} - k + \lfloor j/2 \rfloor - 1.$$ (3.8)

For this case, we define $P(x)$ as

$$P(x) = 1 \iff \Phi_{2k-j}(u, a \oplus d, s, t, b \oplus x, z) = 1.$$ (3.9)

$^3 \oplus$, sum of modulo 2.
Finally, for $x \in \{0, 1\}^{n-2k}$, we define

$$P(x) = 1 \iff x_{2^{2k}+2} \oplus x_{2^{2k}+2} \oplus x_{1} \oplus x_{n-2k}, \quad (3.10)$$

where $x_{2^{2k}+2}, x_{2^{2k}+2}, x_{1},$ and $x_{n-2k} - (2^{2k} + 2)th, (2^{2k} - 2)th, 1st$ and $(n - 2k)th$ letters of $x$, correspondingly.

Thereby, the definition of $P(x)$ is complete.

**Example 2.** If $k = 1$, then $n = 9$.

If $x = x_3x_2x_1x_5x_4x_6x_7x_8x_9$, then

$$P(x_3x_2x_1x_5x_4x_6x_7x_8x_9) = 1 \iff F_2(x_3x_2x_1x_5x_4x_6x_8) = 1.$$

If $x = x_1x_2x_3x_4x_5x_6x_7x_8$, then

$$P(x_1x_2x_3x_4x_5x_6x_7x_8) = 1 \iff \Phi_1(x_1, x_4, x_5x_8) = 1.$$

If $x = x_3x_2x_1x_5x_4x_6x_7$, then

$$P(x_3x_2x_1x_5x_4x_6x_7x_8) = 1 \iff x_1 \oplus x_7 \oplus x_6 \oplus x_5 = 1,$$

where $\Phi_1$ is defined in Example 1 and the definition of $F$ is given in the Appendix.

Since the parts $u, v, s, r, t, w, z$ of word $x$ are located in $x$ symmetrically, the value of $P(x^k)$ will depend on the same parts of word $x$ as $P(x)$. From (3.2), (3.5), and (3.8), one can easily prove that the following proposition is true.

**Proposition 1.** Let $x \in (0, 1)^{n-i}$ and $y \in (0, 1)^{n-j}$, where $0 < i = j < 2k$. Let $x, y$ be represented in the form (3.5) (where $v$ and $w$ are empty in the case $i = 0$ or $j = 0$). Then

1. $s(t)$-parts of $x$ and $y$ have no common occurrences of letters;
2. $s$-parts of words $x'$s for $j = 0, ..., 2k - 1$ contain all letters with numbers from $2k + 1$ to $2^{2k}$;
3. $t$-parts of words $x'$s for $j = 0, 2k - 1$ contain all letters with numbers from $2^{2k} + 1$ to $2^{2k+1}$, except letters with numbers $2^{2k} + 2^{2k-j} + k - 1 - [j/2]$ for $j = 1, ..., 2k - 1$.

Now, we prove

**Lemma 4.** If $x, y \in (0, 1)^{n-2k}$ and $x \neq y$, then $x$ and $y$ are not $n$-equivalent relative to $P(x)$ and $P(x^k)$, respectively.

**Proof.** For the sake of concreteness and without loss of generality, we consider only the case of $P(x)$. Let $x, y \in (0, 1)^{n-2k}$ and $x \neq y$. Let $i$ be the smallest number such that the $i$th letter of $x$ does not equal the $i$th letter of $y$.

Four cases are possible.
COMPLEXITY OF PREDICATES

Case 1. $1 \leq i \leq 2^{2k-1}$. Let $z \in (0, 1)^{2k}$ and let
\[ xz = u_1 v_1 t_1 s_1 z, \quad yz = u_2 v_2 t_2 s_2 z, \]
where $u_1, u_2, s_1, s_2, r_1, r_2, t_1, t_2, z$ satisfy (3.2). From (3.3) of definition $P(x)$ for words of length $n$ we can conclude
\[ P(xz) = P(yz) \iff (F_{2k}(u_1, s_1, t_1, z) = F_{2k}(u_2, s_2, t_2, z)). \]
Since $l(u_1 t_1) = l(u_2 s_2) = 2^{2k-1}$, therefore $u_1 s_1 t_1 \neq u_2 s_2 t_2$. Thus, by Lemma 2, we derive that there exists $z \in (0, 1)^{2k}$ such that
\[ F_{2k}(u_1, s_1, t_1, z) \neq F_{2k}(u_2, s_2, t_2, z). \]
Therefore, $x$ and $y$ cannot be $n$-equivalent relative to $P(x)$.

Case 2. $2^{2k-1} < i \leq n - 2k$ and $i \neq 2^k + 2^{2k-j} + k - 1 - \lfloor j/2 \rfloor$, for $j = 1, \ldots, 2k - 1$. Then, by Proposition 1, there exists a unique $1 \leq j \leq 2k - 1$ such that for any $z \in (0, 1)^{2k-j}$
\[ xz = u_1 v_1 s_1 t_1 w_1 z, \quad uz = u_2 v_2 s_2 t_2 w_2 z, \]
where $u_1, u_2, v_1, v_2, w_1, w_2, s_1, s_2, v_1, v_2, t_1, t_2$ satisfy (3.5) and the $i$th letter of $x(y)$ occurs in $s_j(s_2)$ or $t_j(t_2)$.

From (3.6) and (3.8) of definition $P$ for words of length $n - j$ we can conclude
\[ P(xz) = P(yz) \iff \Phi_{2k-j}(u_1, s_1, t_1, z) = \Phi_{2k-j}(u_2, s_2, t_2, z) \] (3.11)
(in the case of odd $j$), and
\[ P(xz) = P(yz) \iff \Phi_{2k-j}(u_1, a_1 \oplus d_1, s_1, t_1, b_1 \oplus c_1, z) = \Phi_{2k-j}(u_2, a_2 \oplus d_2, s_2, t_2, b_2 \oplus c_2, z) \] (3.12)
(in the case of even $j$).

Since $i > 2k$, $u_1 = u_2$, and $u_1 s_1 t_1 \neq u_2 s_2 t_2 (u_1 a_1 \oplus d_1 s_1 t_1 \neq u_2 a_2 \oplus d_2 s_2 t_2)$ and thus, by Lemma 2, we derive that there exists $z \in (0, 1)^{2k-j}$ such that the right-hand parts of (3.11) and (3.12) do not hold.

Hence, $x$ and $y$ cannot be $n$-equivalent relative to $P(x)$.

Case 3. $i = 2^k + 2^{2k-j} + k - 1 - \lfloor j/2 \rfloor$ for $j = 1, \ldots, 2k - 1$ and for the remaining $i$'s, the $i$th letter of $x$ equals the $i$th letter of $y$. Let us assume that $j$ is even (the case of odd $j$ can be considered analogously). We denote $c_j (c_2)$ as a letter of $x(y)$, whose number equals $i$ and let $d_1 (d_2)$ be a letter of $x(y)$, whose number equals $2^k + 2^{2k-j-1} + k - 1 - \lfloor (j - 1)/2 \rfloor$. Let $z \in (0, 1)^{2k-j}$ and
\[ xz = u_1 v_1 s_1 t_1 w_1 z, \quad yz = u_2 v_2 s_2 t_2 w_2 z, \]
where
\[ u_1, u_2, w_1, w_2, v_1, v_2, s_1, s_2, y_1, t_1, r_1, r_2, z \]
satisfy (3.5). By (3.8) of the definition of the predicate \( P \) for words of length \( n - j \), we can conclude

\[
P(xz) = P(yz) \iff \Phi_{2k-1}(u_1, a_1 \oplus d_1, s_1, t_1, b_1 \oplus c_1, z)
\]

\[
= \Phi_{2k-1}(u_2, a_2 \oplus d_2, s_2, t_2, b_2 \oplus c_2, z),
\]

(3.13)

where \( a_1(b_2) \) and \( b_1(b_2) \) are located in \( xz(yz) \) symmetrically to \( d_1(d_2) \) and \( c_1(c_2) \), respectively. Since \( i > 2k \), \( u_1 = u_2 \), and since numbers of \( a_1(a_2) \) and \( b_1(b_2) \) in \( x \) do not equal \( 2^{2k} + 2^{2k-1-1} + k - 1 - [j/2] \) for any \( j = 1, \ldots, 2k - 1 \) (and therefore, \( a_1 = a_2 \) and \( b_1 = b_2 \)), then \( b_1 \oplus c_1 \neq b_2 \oplus c_2 \) and thus,

\[ u_1(a_1 \oplus d_1)s_1t_1(b_1 \oplus c_1) \neq u_2(a_2 \oplus d_2)s_2t_2(b_2 \oplus c_2) \]

Therefore, by Lemma 3 we derive that there exists \( z \in (0, 1)^{n-j} \) such that the right-hand part of (3.12) does not hold. Hence, \( x, y \) cannot be \( n \)-equivalent relative to \( P(x) \).

**Case 4.** \( i = 2^{2k} + 2 \). This case is completely analogous to Case 3.

The lemma is completely proved.

Now, we want to prove that all words whose lengths are between \( n - 4k + 1 \) and \( n - 2k - 1 \), denoted by the set \( S \), are also non-\( n \)-equivalent relative to \( P(x) \) and \( P(x^R) \).

**Lemma 5.** Any two distinct words \( x, y \) such that

\[
n - 4k + 1 \leq l(x), \quad l(y) < n - 2k
\]

are not \( n \)-equivalent relative to \( P(x) \) and \( P(x^R) \).

**Proof.** Two cases are possible.

**Case 1.** \( l(x) \neq l(y) \). Without loss of generality, we may assume \( l(x) < l(y) \). Let \( h \) be an integer such that \( l(y) + h = n - 2k \). Then, \( l(x) + h < n - 2k \). Since the predicate \( P(x)(P(x^R)) \) equals zero at each word of length \( i \), where \( n - 4k + 1 \leq i \leq n - 2k - 1 \) by the definition of \( P(x) \), then \( P(xz) = 0 \) (\( P((xz)^R) = 0 \)) for any word \( z \) such that \( l(z) = h \). On the other hand, we may always choose a \( z \) such that \( l(z) = h \) and \( P(yz) = 1 \) (and \( P((yz)^R) = 1 \) for some other \( z_1 \)). Hence, it follows that \( x \) and \( y \) are not \( n \)-equivalent relative to \( P(x) \) (and to \( P(x^R) \)).

**Case 2.** \( l(x) = l(y) \). We choose \( h \) such that \( l(x) + h = l(y) + h = n - 2k \). Let \( z \in (0, 1)^h \), then \( l(xz) = l(yz) = n - 2k \) and they are not \( n \)-equivalent relative to \( P(x) \) (and \( P(x^R) \)). Consequently, \( x \) and \( y \) are also not \( n \)-equivalent relative to \( P(x) \) (and \( P(x^R) \)).

The lemma is completely proved.

As a consequence of Lemma 5, we find all words \( x \) such that \( n - 4k + 1 \leq l(x) < n - 2k \) in the set of pairwise non-\( n \)-equivalent words relative to \( P(x) \) and \( P(x^R) \).
The number of such words equals $2^{n-2k} - 2^{n-4k+2}$, and we denote the set of all such words by $S$. By Lemma 5, any two different words $x$ and $y$ such that $l(x) = l(y) = n - 2k$ are not $n$-equivalent relative to $P(x)$ (to $P(x^R)$). However, there exist among all such words those that are $n$-equivalent to words from $(0, 1)^{n-2k-i}$, where $1 \leq i \leq 2k - 1$. Let $r_i$ be the number of words from $(0, 1)^{n-2k}$ which are $n$-equivalent to words from $(0, 1)^{n-2k-i}$.

Our next problem is to estimate $r_i$. Let us start with Proposition 2, which we use for estimating $r_i$.

**Proposition 2.** Let $y \in (0, 1)^{n-2k-i}$, where $1 \leq i \leq 2k - 1$. If $p_1 \neq p_2$ and $p_1, p_2 \in (0, 1)^i$, then for any $z$ such that $0 < l(z) \leq 2k - i$,

$$P(y_{p_1}z) = P(y_{p_2}z) \quad (P((y_{p_1}z)^R) = P((y_{p_2}z)^R)).$$

**Proof.** We consider only the case of $P(x)$, and the case of $P(x^R)$ will be left to the reader. Let $y, p_1, p_2$ satisfy the conditions of the proposition. Then, $l(y_{p_1}z) = l(y_{p_2}z) = n - (2k - l(z))$. We represent $y_{p_1}z$ and $y_{p_2}z$ in the form of (3.4):

$$y_{p_1}z = u_1v_1s_1t_1u_2z, \quad y_{p_2}z = u_2v_2s_2t_2u_2z,$$

where $u_1, u_2, v_1, v_2, s_1, s_2, r_1, r_2, t_1, t_2, c_1, c_2, z$ satisfy (3.5). Since $l(p_1) = l(p_2) = i \leq 2k - 1 \leq 2^{2k-i} - k$, then by (3.6) and (3.9) the values of $P(y_{p_1}z)$ and $P(y_{p_2}z)$ do not depend on the $p_1$- and $p_2$-parts of $y_{p_1}z$ and $y_{p_2}z$, respectively. The proposition is completely proved.

We return now to the problem of estimating $r_i$. By Proposition 2, if $0 \leq i \leq 2k - 1$, then for each $x \in (0, 1)^{n-2k}$ such that there exists an $n$-equivalent $y \in (0, 1)^{n-2k-i}$ relative to $P(x)$ ($P(x^R)$), there may correspond a predicate $R(z)$ for $l(z) \leq 2k - i$ and some word of length $i$.

The number of predicates, which may have different values only for words of length not more than $2k - i$, is easily seen to be not more than $2^{2k-i+1-1}$. Therefore,

$$r_i \leq 2^{2k-i+1-1} \cdot 2^i.$$ 

It follows that the number of words from $(0, 1)^{n-2k}$ which are not pairwise $n$-equivalent relative to $P(x)$ ($P(x^R)$) to words from $S$ is not less than

$$|(0, 1)^{n-2k}| - \sum_{i=1}^{2k-1} r_i \geq 2^{n-2k} - 2^{n-4k+2} + 2^{n-2k} - 2k \cdot 2^{2k} \cdot 2^{2k},$$

and, therefore,

$$p_{T(\omega)}(n) \geq |S| + (0, 1)^{n-2k} - \sum_{i=1}^{2k-1} r_i \geq 2^{n-2k} - 2^{n-4k+2} + 2^{n-2k} - 2k \cdot 2^{2k} \cdot 2^{2k},$$

and, therefore,
where $T(x)$ equals either $P(x)$ or $P(x^R)$. Since $n = n_k = 2^{2k+1} + 2k - 1$, we have $n - 2k + 1 = 2^{2k+1}, 2k + 1 = \log(n - 2k + 1)$. Finally, we obtain 

$$\phi_{T(x)}(n_k) \geq \psi_{T(x)}(n_k) \geq 2^{n_k+2}/n_k$$

for $k \to \infty$, where $T(x) = P(x)$ or $P(x^R)$. This completes the proof of our theorem.

4. Conclusion

We have effectively constructed the predicate $P(x)$, such that $P(x)$ and $P(x^R)$ have maximal possible complexity of $n$-recognition by finite automata and thus is independent of the complexity of $n$-recognition of $P(x)$ on the end from which the word is read by finite automata.

APPENDIX

This appendix is devoted to a proof of Lemma 1. First, we prove

**Lemma A1.** For any $n = 2^k$, $k = 1, 2,\ldots$, a set $G_k$ of $n$ by $n$ symmetric 0-1-valued matrices of cardinality $2^{n-k}$ can be constructed such that every binary $n$-tuple occurs (necessarily uniquely) as a row of some matrix in $G_k$.

**Proof.** Let $n = 2^k$. We define a set $S_k$ of words $x = x_0x_1 \cdots x_{n-1}$ of length $n$ as follows:

$$x \in S_k \iff \text{each digit } x_m, \text{ where } 0 \leq m \leq k - 1 \ (\text{i.e., } x_1, x_2, x_4, x_8, \ldots),$$

is equal to $\sum_{i \in I} x_i$, where $I$ consists of all numbers $i, 0 \leq i \leq n - 1$, except for $2^m$ itself, whose binary representation contains a 1 in the $(m + 1)$ position from the right. For example if $n = 8$ and $m = 0$, then $x_1 = x_3 \oplus x_5 \oplus x_7$; if $m = 1$, then $x_2 = x_3 \oplus x_6 \oplus x_7$; and finally, if $m = 2$, then $x_4 = x_5 \oplus x_6 \oplus x_7$. It can be shown that $x = x_1 \cdots x_n \in S_k$ iff $x = \bar{x}_1 \cdots \bar{x}_n \in S_k$. We note here that $S_k$ is the Hamming code [3].

**Example 1.** $S_2$: $\{0000, 0111, 1000, 1111\}$.

By the definition of $S_k$, the number of different words in $S_k$ equals $2^{n-k}$. We define the set of matrices $G_k = \{D_x\}$, where $D_x$ is a matrix corresponding to $x \in S_k$ as follows:

$$D_x = \begin{bmatrix} x_1 \cdots x_2^k \\ x_1 \cdots x_2^k \\ \vdots \\ x_1 \cdots x_2^k \\ x_1 \cdots x_2^k \end{bmatrix} \oplus \begin{bmatrix} x_1 \cdots x_1 \\ x_1 \cdots x_1 \\ \vdots \\ x_1 \cdots x_1 \\ x_1 \cdots x_1 \end{bmatrix} \oplus \begin{bmatrix} x_1 \cdots x_1 \\ x_1 \cdots x_1 \\ \vdots \\ x_1 \cdots x_1 \\ x_1 \cdots x_1 \end{bmatrix} \oplus \begin{bmatrix} 10 \cdots 0 \\ 10 \cdots 0 \\ \vdots \\ 01 \cdots 0 \\ 00 \cdots 1 \end{bmatrix}, \quad (A1)$$

where each matrix in (A1) is an $n$ by $n$ matrix.

4 If $S$ is a set, then $|S|$ is a cardinality of $S$. 


EXAMPLE 2. We prove that the set $G_k$ satisfies the conditions of the lemma.

Consider $D_x = (a_{ij})$ the matrix corresponding to $x \in S_k$. If $i \neq j$, then $a_{ij} = x_i \oplus x_j \oplus x_j$. Therefore, $D_x$ is symmetric.

It was proved by Hamming [3] that for $x, y \in S_k$ and $1 \leq i, j \leq n$ if $x \neq y$ or $i \neq j$, then

$$x \oplus e_i \neq y \oplus e_j,$$

where $e_i = (0, 0, \ldots, 0, 1, 0, \ldots, 0)$, the one appearing in the $i$th coordinate. Now, for $x, y \in S_k$ and $1 \leq i, j \leq n$, let $a$ be the $i$th row of $D_x$ and $b$ be the $j$th row of $D_y$, i.e.,

$$a = x \oplus (x_1, \ldots, x_i) \oplus (x_i, \ldots, x_i) \oplus e_i,$$

$$b = y \oplus (y_1, \ldots, y_i) \oplus (y_j, \ldots, y_j) \oplus e_i.$$

We want to prove that if $x \neq y$ or $i \neq j$, then $a \neq b$. By definition, $a = x \oplus e_i$ or $a = x' \oplus e_i$, and $b = y \oplus e_j$ or $b = y' \oplus e_j$. But $x \in S_k$ iff $x \in S_k$, so $a = x' \oplus e_i$, where $x' \in S_k$ and $y' = y'$. By (A2), we conclude that $a \neq b$ unless $i = j$ and $x = y'$. Finally, if $i = j$, $x \neq y$, and $x' = y'$, then $x = y$,

$$a = x \oplus (x_1, \ldots, x_i) \oplus (x_i, \ldots, x_i) \oplus e_i = x \oplus (y_1, \ldots, y_i) \oplus (y_j, \ldots, y_j) \oplus e_i = y \oplus (y_1, \ldots, y_i) \oplus (y_j, \ldots, y_j) \oplus e_i = b.$$

Thus, if $x, y \in S_k$ and $x \neq y$ then $D_x \neq D_y$ and moreover, no row appears twice. From the above, $S_k = |G_k| = 2^n - k$. This completes the proof of Lemma A1.

Since the number of matrices in $G_k$ equals $2^n - k$, no row appears twice and each matrix contains $n$ different words as its row, then the matrix set $G_k$ contains exactly $2^{n-k} \cdot n - 2^n$ different words of length $n$ as its rows. Therefore, for any word $x \in (0, 1)^n$ there exists a unique matrix which contains this word as its row.

From the symmetry of each matrix in $G_k$, we can conclude that for any word $x(0, 1)^n$, a unique matrix exists such that it also contains this word as its column.

The matrices of the set $G_k$ will be numbered from 1 to $2^{n-k}$. We now define a Boolean function $F_k$ of $n = n^k - k$ arguments as follows:

$$F_k(x, y, z) = a_{ij}, \quad l(x) = l(x) = k, \quad l(y) = 2^k.$$
\( i = |x| + 1, \ j = |z| + 1, \) and \( a_{ij} \) is an element of the matrix with the number \( |y| + 1 \).\(^5\)

**Example 3.** Let \( k = 2, \ n = 6; \) then the tabulation below represents the values for \( F_2(x_1, x_2, y_1, y_2, z_1, z_2) \). We claim that \( F_k \) satisfies Lemma 1.

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( y_1 )</th>
<th>( y_2 )</th>
<th>( z_1 )</th>
<th>( z_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>00</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>00</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>01</td>
<td>11</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>01</td>
<td>11</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>00</td>
<td>01</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>00</td>
<td>01</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>10</td>
<td>01</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>01</td>
<td>11</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>11</td>
</tr>
</tbody>
</table>

Let \( f(z) \) be a Boolean function of \( k \) variables and let \( \alpha \) be a \( 2^k \)-digit word of its values. Then there exists, by definition of \( F_k \), a unique matrix \( D_{x}(D_{z}) \) such that \( \alpha \) is its row (column). Therefore, there exists \( x(y) \) such that

\[
F_k(x, z) = f(x) \quad \text{and} \quad F(z, y) = f(z).
\]

Lemma 1 is now completely proved.

**Acknowledgments**

The author expresses his deep gratitude to A. Meyer for his very constructive and very helpful discussion of this paper; and he thanks A. Ginzburg and R. Cohen for many useful observations.

\(^5\) If \( y = y_1 \cdots y_r \)-binary word, then \( |y| \) is a number such that \( y \) is its binary representation.
References


