KYBERNETIKA — VOLUME 24 (1988), NUMBER 1

NONDETERMINISM IS ESSENTIAL FOR REVERSAL-BOUNDED TWO-WAY MULTIHEAD FINITE AUTOMATA

ANDREJ BEBJÁK, IVANA ŠTEFÁNEKOVÁ

It is proved for some "nice" function f that $f(n)$-reversal-bounded two-way nondeterministic finite automata are more powerful than $f(n)$-reversal-bounded two-way deterministic ones. This solves a simplified extension of the well-known DLOG ? NLOG problem.

1. INTRODUCTION

The question whether nondeterminism is more powerful than determinism is one of the most investigated questions in complexity theory. The well-known extensions of this question are P ? NP, DLOG ? NLOG. We shall study the latter one in this paper.

We shall consider two-way k-heads deterministic (nondeterministic) automata, $2dfa(k)$ ($2nfa(k)$), because they characterize logarithmic space in the following way

$$\mathsf{DLOG} = \bigcup_{k \in \mathsf{N}} \mathsf{2DFA}(k), \quad \mathsf{NLOG} = \bigcup_{k \in \mathsf{N}} \mathsf{2NFA}(k),$$

where $\mathsf{2DFA}(k)$ ($\mathsf{2NFA}(k)$) is the family of languages recognized by $2dfa(k)$ ($2nfa(k)$) automata.

It was proved in [4] that a specific language can be recognized by no $2nfa(k)$ automaton (for any $k \in \mathsf{N}$) with n^a bound on the number of head reversals in the accepting computations, for $0 < a < \frac{1}{3}$. Using this fact we prove that for some "nice" functions f the class of languages recognized by $f(n)$-reversal-bounded $2nfa(k)$ automata is not closed under complement, and the class of languages recognized by $f(n)$-reversal-bounded $2dfa(k)$ automata is closed under complement. This separates nondeterminism from determinism for reversal-bounded multihead finite automata. The immediate consequence of this fact is that if deterministic multihead finite automata are as powerful as nondeterministic ones then the deterministic automata have to use a substantially larger number of reversals than the nondeterministic automata.

This paper is divided as follows. Section 2 involves some basic definitions and Section 3 contains the basic results concerning "$f(n)$ reversal computability" for multihead finite automata introduced in Section 2. The main results are formulated in Section 4.

2. DEFINITIONS

The formal definition of two-way multihead finite automata as a 5-tuple $(\Sigma, K, F, \delta, q)$ can be found in [6], and it was soon thereafter extensively studied by Sudborough [7], Yao and Rivest [8] and Hromkovič [3].

A bound on the number of reversals as a measure of complexity was introduced by Hartmanis [2]. He considered two-tape Turing machines with one-way read-only input-tape. Reversal-bounded one-tape Turing machines were considered in [1] and reversal-bounded multi-tape Turing machines in [5].

Let f be a function from natural numbers to positive real numbers, and let M denote a family of languages recognized by multihead finite automata from an automaton class $\mathbf{S}$. Then $M - R(f)$ is the class of languages accepted by automata in $\mathbf{S}$ which use in their accepting computations at most $c.f(n)$ head reversals (c is constant, $c \in \mathsf{N}$) for input words of the length n.

Definition 2.1. Let us call $f : \mathsf{N} \to \mathsf{N}$ a *reversal computable* function if there exists a $2dfa(k) - R(f)$ automaton $\mathbf{A}$ such that after finishing computation on the input word of length n the positions of the heads on the input tape represent the value $f(n)$.

In what follows we shall only consider functions for which $f(n) \leqq n$. For the representation of the value $f(n)$ we shall use head H_1, whose position on the ith symbol represents the value i. When proving qualities of the operation O for $f_1, \ldots, f_m$, the result of which is f_{m+1}, we suppose that $f_1, \ldots, f_m, f_{m+1} \leqq n$.

3. REVERSAL COMPUTABLE FUNCTIONS

In this section we shall prove that the sum and integer-valued power of reversal computable functions are reversal computable functions. We will also prove that for some reversal computable functions their integer-valued root is reversal computable.

Lemma 3.1. If f_1, f_2 are reversal computable functions then $f_1 + f_2$ is a reversal computable function, too.

Proof. According to the assumption of the lemma there is a $2dfa(k_1) - R(f_1)$ automaton $\mathbf{A}_1 = (\Sigma_1, K_1, F_1, \delta_1, q_1)$ which constructs the function f_1, and a $2dfa(k_2) - R(f_2)$ automaton $\mathsf{A}_2 = (\Sigma_2, F_2, K_2, \delta_2, q_2)$ which constructs the function f_2, where $K_1 \cap K_2 = \emptyset$. Let $\mathbf{A}_1$ use heads $\mathsf{H}_1^1, \ldots, \mathsf{H}_{k_1}^1$ and $\mathbf{A}_2$ heads $\mathsf{H}_1^2, \ldots, \mathsf{H}_{k_2}^2$.

Let us create a $2dfa(k_1 + k_2 + 3) - R(f_1 + f_2)$ automaton $\boldsymbol{A} = (\Sigma, K_1 \cup K_2 \cup K, F, \delta, q_1)$ which uses heads $H_1^1, \ldots, H_{k_1}^1, H_1^2 \ldots, H_{k_2}^2$, H, F, G. The automaton $\boldsymbol{A}$ works in the following three phases:

1) $\boldsymbol{A}$ simulates work of the automaton $\boldsymbol{A}_1$, at the end of computation the position of head H_1^1 represents the value $f_1(n) = x$.

2) $\boldsymbol{A}$ moves to the initial state of the automaton $\boldsymbol{A}_2$ and simulates it, resulting in head H_1^2 representing the value $f_2(n) = y$.

3) In the last phase of its computation $\boldsymbol{A}$ codes by the position of head H_1^1 the value $x + y$ in the following way: Let heads F and G move by one symbol to the right for every move of H_1^1 to the left until H_1^1 reads left endmarker ¢. F and G then represent the value x. Now, let $\boldsymbol{A}$ move head H_1^1 to the right for every move of G to the left, until G reads ¢. Finally, let $\boldsymbol{A}$ move H_1^2 to the left and simultaneously G, H_1^1 to the right until H_1^2 reads ¢. Head H_1^1 codes by its position on the input tape the value $x + y$. Obviously the whole computation can be done in a more simple way (move H_1^2 to the right for every move H_1^1 to the left until H_1^1 reads ¢), but our goal was to preserve the values x, y that we shall make use of in the following constructions. In the introduced construction of the automaton $\boldsymbol{A}$ its heads execute $O(x) + O(y) + 5 = O(x + y)$ reversals. □

Consequence 3.2. If f is a reversal computable function, then $c.f$ $(c \in \mathbb{N})$ is a reversal computable function as well.

Lemma 3.3. If f is a reversal computable function, then f^i, where $i \in \mathbb{N}$, is a reversal computable function.

Proof. By induction on the exponent i. For $i = 1$ the statement is obvious. Let the statement hold for $i = j$. We shall prove that it holds for $i = j + 1$ as well. According to the induction assumption there is a $2dfa(k_1) - R(f^j)$ automaton $\boldsymbol{A}_1$, which constructs the function f^j, and according to the assumption of Lemma 3.3. there is a $2dfa(k_2) - R(f)$ automaton $\boldsymbol{A}_2$, which constructs the function f. Coming up from equation

$$f^{j+1}(n) = f^j(n) \,.\, f(n) = \underbrace{f^j(n) + f^j(n) + \ldots + f^j(n)}_{f(n)\text{-times}}$$

we create an automaton $\boldsymbol{A}$, which will represent, for the input of length n, at the end of computation by position of its head the value $f^{j+1}(n)$. The automaton $\boldsymbol{A}$ gradually simulates the work of the automaton $\boldsymbol{A}_1$ (during simulation it executes $O(f^j(n))$ reversals) and the work of the automaton $\boldsymbol{A}_2$ (during simulation it executes $O(f(n))$ reversals). In the last phase the automaton $\boldsymbol{A}$ $f(n)$-times adds $f^j(n)$. Addition of two numbers (regardless of their size) requires a constant number of reversals. The number of reversals executed during $f(n)$-times repeated addition will be therefore

$O(f(n))$. The whole number of additions executed by the automaton $\boldsymbol{A}$ is $O(f^j(n)) + O(f(n)) + O(f(n)) = O(f^j(n))$. □

Consequence 3.4. For a reversal computable function f it holds, that there is a $2dfa(k) - R(f)$ automaton, which constructs a function $f^i(n)$, for $i \in \mathsf{N}$.

Lemma 3.5. The function $\lfloor n^{1/i} \rfloor$ is reversal computable.

Proof. We construct a $2dfa(k) - R(\lfloor n^{1/i} \rfloor)$ automaton $\boldsymbol{A}$, which will gradually verify whether $1^i = n?$, $2^i = n?$, etc. If it holds for number j that $j^i \leqq n$ and $(j+1)^i > n$, then obviously $j = \lfloor n^{1/i} \rfloor$. Let us describe the operation of the automaton $\boldsymbol{A}$ in a nonformal way. The automaton $\boldsymbol{A}$ uses heads $\mathsf{G}, \mathsf{H}_1, \mathsf{H}_2, \ldots, \mathsf{H}_i, \ldots, \mathsf{H}_k$. Head G will gradually represent by its position on the input word the values $1^i, 2^i, 3^i, \ldots$ until it reads the right endmarker \$. Head H_1 moves one symbol to the right after each successful cycle (i.e. transition from configuration in which head G represents the value x^i to the configuration in which head G represents the value $(x+1)^i$). At the end of computation of the automaton $\boldsymbol{A}$ head H_1 represents the value $\lfloor n^{1/i} \rfloor$. On transition from configuration in which head G represents the value x^i to configuration in which G represents $(x+1)^i$ heads $\mathsf{H}_2, \ldots, \mathsf{H}_i$ always represent the following information

$$\mathsf{H}_2 \text{ represents the value } \binom{i}{1} \cdot x^{i-1}$$

$$\mathsf{H}_3 \text{ represents the value } \binom{i}{2} \cdot x^{i-2}$$

$$\mathsf{H}_i \text{ represents the value } \binom{i}{i-1} \cdot x .$$

According to the binomial theorem it holds:

$$(x+1)^i = x^i + \binom{i}{1} \cdot x^{i-1} + \ldots + \binom{i}{i-1} \cdot x + 1 .$$

Thus, it is sufficient to move gradually head G to the right by the values represented by heads $\mathsf{H}_2, \ldots, \mathsf{H}_i$ and to move it by one more symbol (each move is done only if the right endmarker is not being read). Each of the heads $\mathsf{H}_2, \ldots, \mathsf{H}_i$ has its own substitute which in the case of value addition keeps information represented by the head. The number of heads is constant and every addition requires a constant number of head reversals. If all addition were done, then head G represents the value $(x+1)^i$ and head H_1 moves by one symbol to the right. The automaton must then adapt information carried by heads $\mathsf{H}_2, \ldots, \mathsf{H}_i$ to the values $\binom{i}{1} \cdot (x+1)^{i-1}, \binom{i}{2} \cdot (x+1)^{i-2}, \ldots, \binom{i}{i-1} \cdot (x+1)$, which are necessary to realize the following cycle.

For above given values the following relations hold:

$$\binom{i}{1}(x+1)^{i-1} =$$

$$= \binom{i}{1}\left(x^{i-1} + \binom{i-1}{1}x^{i-2} + \binom{i-1}{2}x^{i-3} + \ldots + \binom{i-1}{i-2}x + 1\right) =$$

$$= \underbrace{\binom{i}{1}x^{i-1}}_{H_2} + \underbrace{\binom{i}{1}\binom{i-1}{1}x^{i-2}}_{2 \times H_3} + \underbrace{\binom{i}{1}\binom{i-1}{2}x^{i-3}}_{3 \times H_4} + \ldots + \binom{i}{1}$$

$$\binom{i}{2}(x+1)^{i-2} =$$

$$= \underbrace{\binom{i}{2}x^{i-2}}_{H_3} + \underbrace{\binom{i}{2}\binom{i-2}{1}x^{i-3}}_{3 \times H_4} + \underbrace{\binom{i}{2}\binom{i-2}{2}x^{i-4}}_{6 \times H_5} + \ldots +$$

$$+ \underbrace{\binom{i}{2}\binom{i-2}{i-3}}_{(i-1)(i-2)/2 \times H_i} x + \binom{i}{2}$$

etc.

The whole adaptation of information at every cycle from x^i to $(x+1)^i$ always requires constant number of reversals.

Since the maximum number of successful cycles is $\lfloor n^{1/i} \rfloor$, the number of reversals done by the automaton heads is $c \cdot \lfloor n^{1/i} \rfloor + d = O(\lfloor n^{1/i} \rfloor)$, where d is constant number of reversals in case that $\lfloor n^{1/i} \rfloor < n^{1/i}$ and automaton tries to complete one more cycle. □

Example 3.6. The function $\lfloor \log_2 n \rfloor$ is reversal computable. A $2dfa(4) - R(\lfloor \log_2 n \rfloor)$ automaton $\boldsymbol{A}$ representing at the end of computation the value $\log_2 n$ by head H_1 works as follows: head H_2 represents the number 2^x, heads H_3 and H_4 represent alternatively 2^x and the initial position. Transition from configuration in which H_2 represents the value 2^x to the configuration in which H_2 represents $2 \cdot 2^x$ runs like this: Let H_2, H_3 represent 2^x and head H_4 reads %, then in each step H_3 moves by one symbol to the left, H_2 by one symbol to the right and H_4 by two symbols to the right. If H_2 reads \$ in none of the steps, then it represents 2^{x+1}, H_4 represents 2^{x+1} as well and H_3 reads %.

After this successful cycle H_1 moves by one symbol to the right. This is repeated up to the point, when H_2 reads \$. Then the position of H_1 represents $\lfloor \log_2 n \rfloor$.

Consequence 3.7. The function $\lfloor n^{1/p} \rfloor^q$ $p, q \in \mathbb{N}$, $p \geqq q$ is reversal computable.

4. DETERMINISM VERSUS NONDETERMINISM FOR REVERSAL-BOUNDED MULTIHEAD FINITE AUTOMATA

Now, proving that nondeterminism is essential for reversal-bounded two-way multihead finite automata we give a partial answer to the well-known open problem whether two-way nondeterministic multihead finite automata are more powerful than deterministic ones.

Theorem 4.1. Fo every reversal computable function $f(n)$ the class of languages $\bigcup_{k\in N} 2DFA(k) - R(f)$ is closed under complement.

Proof. It is sufficient to prove that for every language $\mathbf{L}$ in $\bigcup_{k\in N} 2DFA(k) - R(f)$ recognized by a $2dfa(k) - R(f)$ automaton $\mathbf{A}$, $\mathbf{A} = (\Sigma, K, F, \delta, q_0)$, there is a $2dfa(k') - R(f)$ automaton $\mathbf{A}' = (\Sigma, K', F', \delta', q_0')$ recognizing language $\mathbf{L}^c$.

The automaton $\mathbf{A}'$ works on the input word of length n as follows: In the first phase the value $f(n)$ is coded by the position of one of its heads. Let us call this head G. To code this value at most $c_i \cdot f(n)$ reversals are necessary since $f(n)$ is a reversal computable function. Then $\mathbf{A}'$ moves to the initial state q_0 of the automaton $\mathbf{A}$. In the second phase of the computation $\mathbf{A}'$ simulates computation of the automaton $\mathbf{A}$. For every reversal of one of the heads of the automaton $\mathbf{A}$ automaton $\mathbf{A}'$ moves the head G one symbol to the left. During the computation only one of the following three cases is possible:

1) The automaton $\mathbf{A}$ would enter the state $q \in F$. Then $\mathbf{A}'$ does not accept and enters a state $p \notin F'$.

2) If head G of the automaton $\mathbf{A}'$ reads $\not{c}$, then the automaton $\mathbf{A}$ would already cross bound of reversals and therefore automaton $\mathbf{A}'$ enters a state $p \in F'$.

3) The head G of the automaton $\mathbf{A}'$ does not read yet $\not{c}$ and for state p, in which automaton $\mathbf{A}$ would be, δ-function is not defined. In such case automaton $\mathbf{A}'$ accepts.

Since we simulated the work of the automaton $\mathbf{A}$, the number of reversals does not (even in this part of computation of $\mathbf{A}'$) exceed the value $c_2 \cdot f(n)$, where c_2 is a constant. Hence $\mathbf{A}'$ is a $2dfa(k') - R(f)$ automaton and recognizes the language $\mathbf{L}^c$. □

Theorem 4.2. For all a: $0 < a < \frac{1}{3}$, the class of languages $\bigcup_{k\in N} 2NFA(k) - R(n^a)$ is not closed under complement.

Proof. Let us consider the language $\mathbf{L}$, for which it is proved in [4], that $\mathbf{L}$ does not belong to $\bigcup_{k\in N} 2NFA(k) - R(n^a)$, where $0 < a < \frac{1}{3}$. It is sufficient to prove that the complement of the language $\mathbf{L} = \{x_1 \# x_2 \# \ldots x_k \# \mid k \geqq 0,\ x_i \in \mathbf{L}_R$ for $i = 1, 2, \ldots, k\}$, where $\mathbf{L}_R = \bigcup_{r\in N} \mathbf{L}_r$,

$$\mathbf{L}_r = \{w_1 c w_2 c \ldots c w_r + w_r c \ldots c w_2 c w_1 \mid w_i \in \{0, 1\}^* \text{ for } i = 1, 2, \ldots, r\}$$

is recognized by a $2nfa(2) - R(2)$ automaton $\mathbf{A}$.

Let us analyze in more detail, what is the structure of the words of the language

$\mathbf{L}^c$. Since every word in $\mathbf{L}$ (besides ε) contains the symbol #, we conclude that $W_1 = \{0, 1, c, +\}^+ \subset \mathbf{L}^c$. The language contains further all words from W_2 and W_3, $W_2 = \{w \mid w = \# v,\ v \in \{0,1,c,+,\#\}^*\}$, $W_3 = \{w \mid w = vx,\ v \in \{0, 1, c, +, \#\}^*$ $x \in \{0, 1, c, +\}\}$. All other words of the language $\mathbf{L}^c$ (i.e. those which do belong neither to W_1, W_2 nor to W_3) have the following structure:

$$w = x_1 \# x_2 \# \ldots \# x_k \#, \quad \text{where} \quad k \geqq 1, \quad x_i \in \{0, 1, c, +\}^*$$
$$\text{for} \quad i = 1, \ldots, k \quad x_1 \neq \varepsilon \quad \text{and there is } i \text{ such that} \quad x_i \notin \mathbf{L}_\mathbf{R}.$$

The automation $\boldsymbol{A}$ nondeterministicly decides at the beginning of the computation on word u, which one of the four above mentioned structure is acquired by the word u. To verify the first three cases one head is sufficient and this one executes not even one reversal. In the fourth case the automaton again nondeterministicly decides which one of the subwords x_i does not belong to $\mathbf{L}_\mathbf{R}$. To verify its decision it is sufficient, for the automaton, to use two heads that execute two reversals. □

Theorem 4.3. For any reversal computable function f such that $f(n) \leqq n^a$, where $0 < a < \frac{1}{3}$

$$\bigcup_{k \in \mathsf{N}} \mathsf{2DFA}(k) - R(f) \subsetneqq \bigcup_{k \in \mathsf{N}} \mathsf{2NFA}(k) - R(f).$$

Consequence 4.4. For $p, q \in \mathsf{N}$, $0 < p/q < \frac{1}{3}$

$$\bigcup_{k \in \mathsf{N}} \mathsf{2DFA}(k) - R(\lfloor n^{1/q} \rfloor^p) \subsetneqq \bigcup_{k \in \mathsf{N}} \mathsf{2NFA}(k) - R(\lfloor n^{1/q} \rfloor^p).$$

Consequence 4.5. For $p \in \mathsf{N}$

$$\bigcup_{k \in \mathsf{N}} \mathsf{2DFA}(k) - R(\lfloor \log_2 n \rfloor^p) \subsetneqq \bigcup_{k \in \mathsf{N}} \mathsf{2NFA}(k) - R(\lfloor \log_2 n \rfloor^p).$$

(Received August 11, 1986.)

REFERENCES

[1] P. C. Fischer: The reduction of tape reversals for off-line one-tape Turing machines. J. Comput. System Sci. *2* (1968), 136—147.

[2] J. Hartmanis: Tape reversal bounded Turing machines computations. J. Comput. System Sci. *19* (1979), 145—162.

[3] J. Hromkovič: One-way multihead deterministic finite automata. Acta Inform. *19* (1983), 377—384.

[4] J. Hromkovič: Fooling a two-way nondeterministic multihead automaton with reversal number restriction. Acta Inform. *22* (1985), 589—594.

[5] T. Kameda and R. Vollmar: Note on tape-reversal complexity of languages. Inform. and Control *17* (1970), 203—215.

[6] T. F. Piatkowski: N-head Finite State Machines. Ph. D. Dissertation. University of Michigan 1963.

[7] I. H. Sudborough: One-way multihead writing finite automata. Inform. and Control *30* (1976), 1—20.

[8] A. C. Yao and R. L. Rivest: K + 1 heads are better than K. J. Assoc. Comput. Mach. *25* (1978), 337—340.

RNDr. Andrej Bebják, RNDr. Ivana Štefáneková, Katedra teoretickej kybernetiky Matematicko-fyzikálnej fakulty Univerzity Komenského (Department of Theoretical Cybernetics — Comenius University), 842 15 Bratislava. Czechoslovakia.