Electronic Colloquium on Computational Complexity, Report No. 92 (2012)

A Note On the Hierarchy of One-way Data-Independent Multi-Head Finite Automata

Pavol Ďuriš [1]

Department of Informatics
Faculty of Mathematics, Physics and Informatics
Comenius University
Mlynská dolina
842 48 Bratislava
Slovakia

Abstract In this paper we deal with one-way multi-head data-independent finite automata. A k-head finite automaton A is data-independent, if the position of every head i after step t in the computation on an input w is a function that depends only on the length of the input w, on i and on t (i.e. the trajectories of heads must be the same on the inputs of the same length). It is known that $k(k+1)/2+4$ heads are better than k for one-way k-head data-independent finite automata. We improve here this result by showing that $2k+2$ heads are better than $\sqrt{2}k$ heads for such automata.

Keywords: computational power, one-way multihead automata, data-independent automata

[1] This work was supported by Slovak Grant Agency for Science (VEGA) under contract #1/0726/09 "Algorithmics and Complexity Aspects of Information Processing".

ISSN 1433-8092

1 Introduction

In [1], [2], Holzer investigates data-independent multi-head finite automata. A k-head finite automaton A is data-independent, if the position of every head i after step t in the computation on an input w is a function $f_A(|w|, i, t)$ (i.e. the trajectories of heads must be the same on the inputs of the same length). One can show for one-way as well as for two-way data-independent k-head finite automata, that determinism is as powerful as nondeterminism for such automata [2], [1], since the only nondeterminism left in the k-head nondeterministic data-independent finite automata is the way in which the next state is chosen. It is also known that $k+1$ heads are more powerful than k for two-way data-independent finite automata [3]. This result follows from an analogical result for deterministic two-way k-head finite automata that holds for unary languages [4]; one has to realize that each deterministic two-way k-head finite automaton behaves on unary inputs as a deterministic data-independent two-way k-head finite automaton.

Surprisingly, it is unknown how strict is the one-way k-head hierarchy for data-independent finite automata. In [3] it is shown that $k(k+1)/2+4$ heads are better than k for one-way k-head data-independent finite automata. We improve here this result by showing that $2k+2$ heads are better than $\lfloor\sqrt{2}k\rfloor$ heads for such automata.

Note that it is also known that $k+1$ heads are more powerful than k for one-way k-head finite automata [5].

More results on multi-head data-independent finite automata can be found in [3], and [1], where also some results on such nonuniform automata are presented.

2 Preliminaries

By $|S|$ we denote cardinality of a set S and by $|x|$ we denote the length of a string x.

A k-head finite automaton A is data-independent, if the position of every head i after step t in the computation on an input w is a function $f_A(|w|, i, t)$ (i.e. the trajectories of heads must be the same on the inputs of the same length).

For $k \geq 1$ let k-DiDFA denote deterministic data-independent finite automaton with k one-way input heads. We assume that the input is augmented with two endmarkers \$ (one at the beginning and the second one at the end of the input), and each head scans the left endmarker at the beginning of the computation.

Let A be a k-DiDFA. A configuration of A is $(k+1)$-tuple $(p_1, p_2, \ldots, p_k, q)$, where p_i is the position of the i-th head from the left-end of the input tape, and q denotes the current state.

By $\mathcal{L}(k\text{-DiDFA})$ we denote the family of languages accepted by k-DiDFAs.

3 Results

Yao and Rivest show in [5] that for every $k \geq 1$ there is a language R_k that can be recognized by a one-way $(k+1)$-head deterministic finite automaton, but it cannot be recognized by any one-way k-head nondeterministic finite automaton. Being inspired by this result, Holzer, Kutrib and Malcher show the following result in [3].

Theorem 1 [3]. *Let $k \geq 2$. Then $\mathcal{L}(k\text{-DiDFA}) \subset \mathcal{L}((k(k+1)/2+4)\text{-DiDFA})$.*

Here we improve this result by showing that $2k+2$ heads are better than $\lfloor\sqrt{2}k\rfloor$ heads for one-way data-independent finite automata. To do so, we need to define a language L_k for every $k \geq 1$.

Before defining L_k, let us consider the following simple example with the four strings w_0, w_1, w_2 and w_3: Let $w_1 = 100111010$, $w_2 = 001011101$ and $w_3 = 111010001$, and let

$$w_0 = \begin{bmatrix} 100111010 \\ 001011101 \\ 111010001 \end{bmatrix}.$$

To encode w_0 with 3 tracks, formally, we can use an alphabet with $2^3 = 8$ symbols; hence, $|w_0| = 9 = |w_1| = |w_2| = |w_3|$. In such a case, we say that the i-th track of w_0 is w_i for $i = 1, 2, 3$. We generalize this idea to define L_k as follows.

For each integer $k \geq 1$, let Σ_k be an alphabet with $|\Sigma_k| = 2^{k^2}$ and $\Sigma_k \cap \{0,1\} = \emptyset$, and let L_k be the language (over the alphabet $\Sigma_k \cup \{0,1\}$) containing all strings w of the form $w = w_0 w_1 w_2 \ldots w_{k^2}$, where $|w_0| = |w_i|$

and $w_i \in \{0,1\}^*$ for $i = 1, 2, \ldots, k^2$, $w_0 \in \Sigma_k^*$ and the i-th track of w_0 is w_i for $i = 1, 2, \ldots, k^2$.

Theorem 2. *Let $k \geq 1$. Then $L_k \in \mathcal{L}((2k+2)$-DiDFA)*

Proof. L_k can be accepted by a $(2k+2)$-DiDFA B, that recognizes the inputs in phases starting by a preparing phase, which is followed by k comparing phases.

During the preparing phase, the $(2k+2)$th head of B traverses the whole input w by the maximal speed, (i.e. it moves to the right at every step). This phase ends, when this head enters the right endmarker. This head also checks whether $|w| = (1+k^2)m$ for some nonnegative integer m. If not, then B reject the input. Now assume that $|w| = (1+k^2)m$ for some m, and let $w = w_0 w_1 \ldots w_{k^2}$, where $|w_i| = m$ for each i. In such a case, the preparing phase takes $(1+k^2)m + 1$ steps. Moreover, the first k heads are distributed so that the first, the second, the third,..., the kth head enters w_1, w_{k+1}, w_{2k+1},..., $w_{(k-1)k+1}$, respectively, at the end of this phase. Such distribution can be performed, if the ith head $(1 \leq i \leq k)$ crosses $(i-1)k+1$ tape cells during every $1+k^2$ steps, and hence, it crosses the prefix $w_0 w_1 \ldots w_{(i-1)k}$ during $(1+k^2)m+1$ steps of this phase. By some technical reasons, the $(2k+1)$th head moves by the same way as the first one during this phase.

During the first comparing phase, the $(2k+1)$th head crosses the whole suffix $w_1 w_2 \ldots w_{k^2}$ by the maximal speed, and this phase ends, when this head enters the right endmarker. This phase takes $k^2 m$ steps. Moreover, the ith head $(1 \leq i \leq k)$ traverses the string $w_{(i-1)k+1}$ and the $(k+1)$th head traverses w_0 by the same speed crossing exactly one tape cell during every k^2 steps. Hence, each head moving by such speed crosses only one w_l during this phase. This enables B to compare the string $w_{(i-1)k+1}$ (traversed by the ith head) with the corresponding track of w_0 (traversed by the $(k+1)$th head) for each i with $1 \leq i \leq k$.

The jth comparing phase, $(2 \leq j \leq k)$, is very similar to the first comparing phase with the exception that the $(2k+1)$th and the $(k+1)$th head are replaced by the $(k+j-1)$th and by the $(k+j)$th head, respectively. Note that, the ith head $(1 \leq i \leq k)$ traverses $w_{(i-1)k+j}$ and the $(k+j)$th head traverses w_0 during this phase. □

Theorem 3. *Let $k \geq 1$. Then $L_k \notin \mathcal{L}(l$-DiDFA), if $\binom{l}{2} < k^2$.*

Our proof imitates Yao and Riverst's proof that a certain language cannot be accepted by any one-way k-head finite automaton [5]. Using some counting arguments and the assumption that our automaton, (that should recognize L_k), has only l heads, where $l(l-1)/2 < k^2$, we will show that there are two different strings $\bar{w} = z_1\bar{\delta}_0^s z_2 \bar{\delta}_r^s z_3$ and $\tilde{w} = z_1\tilde{\delta}_0^s z_2 \tilde{\delta}_r^s z_3$ in L_k with the same pattern of behavior of heads during the corresponding accepting computations. Moreover, $\bar{\delta}_0^s$ and $\bar{\delta}_r^s$ (and $\tilde{\delta}_0^s$ and $\tilde{\delta}_r^s$) are never scanned simultaneously during the corresponding accepting computations. This will enable us to show that the mixed string $\hat{w} = z_1\bar{\delta}_0^s z_2 \tilde{\delta}_r^s z_3$, which is not in L_k, will be accepted. (Note that our proof does not use the fact that our automaton is data-independent one, and hence, we have that L_k cannot be recognized by any one-way l-head finite automaton with $l(l-1)/2 < k^2$.)

Proof. Assume to the contrary that there is a l-DiDFA A accepting L_k, where

$$\binom{l}{2} < k^2. \tag{1}$$

Let

$$d = \binom{l}{2} k^2 + 1. \tag{2}$$

Let Q be set of states of A. Choose n so that the following inequality holds.

$$(|Q|((k^2+1)dn+2)^l)^{2l+3} < 2^n/(k^2 d). \tag{3}$$

Let L_k^n be the language containing all the strings of the length $(k^2+1)dn$ from L_k. Clearly, each string w in L_k^n can be written in the form $w = w_0 w_1 w_2 \ldots w_{k^2}$, where each $w_i = \delta_i^1 \delta_i^2 \delta_i^3 \ldots \delta_i^d$ for some δ_i^j's, and $|\delta_i^j| = n$ for every $i = 0, 1, 2, \ldots, k^2$ and $j = 1, 2, 3, \ldots, d$.

Let w be any string from L_k^n with the corresponding substrings δ_i^j's as above, and let u and v be any two heads of A and let (i,j) be any ordered pair of two integers with $1 \le i \le k^2$ and $1 \le j \le d$. We will say that the ordered pair (u,v) of heads of A *covers* the ordered pair (i,j) of integers in the computation of A on w, if there is a time step t of that computation at which the head u scans the string δ_0^j and at which the head v scans the string δ_i^j. (For convenience we use the notion *pair* instead of *ordered pair* in the rest of this proof.)

Now our aim is to prove the following lemma.

Lemma 1. *For each w in L_k^n there is a pair (p, q) of integers (with $1 \leq p \leq k^2$ and $1 \leq q \leq d$) which is not covered by any pair of heads in the computation of A on w, (i.e. the substrings δ_0^q and δ_p^q of w are never scanned simultaneously during the computation of A on w).*

Proof. Let w be any string from L_k^n with the corresponding substrings w_i's and δ_i^j's as above. Let (u, v) be any pair of heads of A and let $h_{u,v}$ denote the number of pairs of integers (i, j) covered by the pair (u, v) in the computation of A on w.

Firstly, we will show that $h_{u,v} \leq d + k^2$, (i.e. the pair (u, v) of heads can cover at most $d + k^2$ pairs (i, j) of integers in the computation of A on w). For every $i = 1, 2, 3, \ldots, k^2$, let B_i be the set of integers j with $1 \leq j \leq d$ such that (u, v) covers (i, j) in the computation of A on w. One can easily observe that

$$h_{u,v} = \sum_{i=1}^{k^2} |B_i|. \tag{4}$$

Now assume that $B_m \neq \emptyset$ and $B_{m'} \neq \emptyset$ for some m, m' with $1 \leq m < m' \leq k^2$, and assume that $j \in B_m$ and $j' \in B_{m'}$ for some j, j' with $1 \leq j, j' \leq d$. This means that there are time steps t and t' of the computation of A on w such that the head u scans δ_0^j and the head v scans δ_m^j at the time t, and similarly, the head u scans $\delta_0^{j'}$ and the head v scans $\delta_{m'}^{j'}$ at the time t'. Since $1 \leq m < m' \leq k^2$ (see the assumption above), we can write w in the form $w = w_0 x_1 \delta_m^j x_2 \delta_{m'}^{j'} x_3$ for some strings x_1, x_2, x_3. Moreover, v is the one-way head, and therefore v cannot enter $\delta_{m'}^{j'}$ before leaving δ_m^j. Thus, $t < t'$. This means that $j \leq j'$, since otherwise w_0 would be of the form $w_0 = y_1 \delta_0^{j'} y_2 \delta_0^j y_3$ for some strings y_1, y_2, y_3, but it would contradict the fact above that the one-way head u scans δ_0^j at the time t and u scans $\delta_0^{j'}$ at the time t', where $t < t'$. The fact $j \leq j'$ shown above yields that

$$\max B_m \leq \min B_{m'} \text{ for } 1 \leq m < m' \leq k^2 \text{ with } B_m \neq \emptyset \neq B_{m'}. \tag{5}$$

For every $i = 1, 2, 3, \ldots, k^2$, let D_i be set obtained from B_i by deleting the maximal element from B_i, if $B_i \neq \emptyset$, and $D_i := \emptyset$, if $B_i = \emptyset$. Clearly,

$$|B_i| \leq |D_i| + 1 \text{ for every } i = 1, 2, 3, \ldots, k^2. \tag{6}$$

By (5) and by the construction of D_i's above, we have that

$$D_m \cap D_{m'} = \emptyset \text{ for } 1 \leq m < m' \leq k^2. \tag{7}$$

By (4), (6), (7), and by the fact that $D_i \subseteq B_i \subseteq \{1, 2, 3, \ldots, d\}$, for $i = 1, 2, \ldots, k^2$, (see above), we have that

$$h_{u,v} = \sum_{i=1}^{k^2} |B_i| \leq \sum_{i=1}^{k^2} |D_i| + k^2 = |\bigcup_{i=1}^{k^2} D_i| + k^2 \leq d + k^2. \tag{8}$$

Now we are ready to find the desired pair (p, q) of integers for w as follows. Let t_u [let t_v] be the time step of the computation of A on w at which the head u [the head v] enters the substring w_1 of w. If $t_u < t_v$, then there is no time step of the computation of A on w, at which u scans some δ_0^j with $1 \leq j \leq d$ (i.e. u scans the substring w_0) and at which v scans some δ_i^j with $1 \leq i \leq k^2$ (i.e. v scans the substring $w_1 w_2 \ldots w_{k^2}$), since both heads u and v scan $\$w_0$ before time t_u, u scans the substring $w_1 w_2 \ldots w_{k^2}\$$ and v scans $\$w_0$ during the time interval $< t_u, t_v - 1 >$, and both heads u and v scan the substring $w_1 w_2 \ldots w_{k^2}\$$ from the time t_v. It means that $h_{u,v} = 0$, if $t_u < t_v$. Similarly, $h_{v,u} = 0$, if $t_u > t_v$, and $h_{u,v} = h_{v,u} = 0$, if $t_u = t_v$. Moreover, $h_{u,v} = 0$, if $u = v$, since the same head cannot scan the substring w_0 and also the substring $w_1 w_2 \ldots w_{k^2}$ at the same time. These results yield that the number of $h_{u,v}$'s with $h_{u,v} > 0$ is at most $\binom{l}{2}$. Consequently, the number of all pairs (i, j) of integers covered by all pairs of heads of A in the computation of A on w is at most $\binom{l}{2}(d + k^2)$, see (8). But the number of all (i, j)'s with $1 \leq i \leq k^2$ and $1 \leq j \leq d$ is

$$k^2 d > \binom{l}{2}(d + k^2), \tag{9}$$

by (1) and (2). Thus, (9) guarantees the existence of the desired pair (p, q) of integers, which is not covered by any pair of heads in the computation of A on w. $\square$

Now our aim is to find the strings $\bar{w}$ and $\tilde{w}$ mentioned above in the idea of the proof. We use some counting arguments to find them. Then we derive a contradiction by showing that a mixed string $\hat{w} \notin L_k$, constructed from $\bar{w}$ and $\tilde{w}$, will be accepted by A.

For each pair of integers (p, q) with $1 \leq p \leq k^2$ and $1 \leq q \leq d$, let $S_{p,q}$ be the set of all strings in L_k^n such that the pair (p, q) is not covered by any pair of heads in the corresponding computations of A on these strings. Since the number of such pairs (p, q) is $k^2 d$, since $|L_k^n| = 2^{k^2 dn}$, (each w_i with $i \geq 1$ can be chosen arbitrarily and w_0 is determined by $w_1, w_2, \ldots, w_{k^2}$), and since each w in L_k^n belongs to some $S_{p,q}$, (by Lemma 1), then there is a pair (r, s) of integers with

$$2^{k^2 dn}/(k^2 d) = |L_k^n|/(k^2 d) \leq |S_{r,s}|. \tag{10}$$

Let w be any string in $S_{r,s}$ with the corresponding substrings δ_j^j's as above. Consider the sequence of configurations in the accepting computation of A on w:

$C_0 \vdash C_1 \vdash \cdots \vdash C_t$.

An occurrence of C_i is said to be *important* for w, it at this step any head enters δ_0^s or δ_r^s. Let $C_{i_1}, C_{i_2}, \ldots, C_{i_g}$ be the sequence of all configurations in the sequence above, that are important for w, let $C_{i_0} = C_0$ and let $C_{i_{g+1}}$ be the accepting configuration in the sequence above. We will say that the sequence $C_{i_0}, C_{i_1}, C_{i_2}, \ldots, C_{i_g}, C_{i_{g+1}}$ is a profile of w. Since each head of A can enter δ_0^s [can enter δ_r^s] during the computation of A on w at most one time, then we have $g \leq 2l$. Thus the number of all different profiles of all strings in $S_{r,s}$ is at most $(|Q|((k^2+1)dn+2)^l)^{2l+3}$ and by (3), it is less than $2^n/(k^2 d)$.

Let w be any string in $S_{r,s}$, where $w = w_0 w_1 \ldots w_{k^2}$, $w_i = \delta_i^1 \delta_i^2 \ldots \delta_i^d$ and $|\delta_i^j| = n$ for $i = 0, 1, 2, \ldots, k^2$ and $j = 1, 2, \ldots, d$. Let the (r, s) *deletion* of the string w be the string obtained from w by deleting δ_0^s and δ_r^s from it. Thus each string w in $S_{r,s}$ can be written in the form $w = z_1 \delta_0^s z_2 \delta_r^s z_3$, where $z_1 z_2 z_3$ is the corresponding (r, s) deletion of w. The number of distinct (r, s) deletions of all strings in L_k^n is $2^{(k^2 d - 1)n}$, since each δ_i^j with $i \geq 1$ (with the exception for δ_r^s) can be chosen arbitrarily and each δ_0^j with $j \neq s$ is determined by the substrings $\delta_1^j, \delta_2^j, \ldots, \delta_{k^2}^j$ being chosen arbitrarily. Sine $S_{r,s} \subseteq L_k^n$, then the number of distinct (r, s) deletions of all strings in $S_{r,s}$ is at most $2^{(k^2 d - 1)n}$.

By (10), by the fact that the number of all different profiles of all strings in $S_{r,s}$ is less than $2^n/(k^2 d)$ (see above), and by the fact that the number of distinct (r, s) deletions of all strings in $S_{r,s}$ is at most $2^{(k^2 d - 1)n}$, (see above), we have that there are two different strings $\bar{w} = z_1\bar{\delta}_0^s z_2\bar{\delta}_r^s z_3$ and $\tilde{w} = z_1\tilde{\delta}_0^s z_2\tilde{\delta}_r^s z_3$ in $S_{r,s}$ (for some $z_1, z_2, z_3, \bar{\delta}_0^s, \bar{\delta}_r^s, \tilde{\delta}_0^s, \tilde{\delta}_r^s$) with the same (r,s) deletion $z_1z_2z_3$ and with the same profile $C_{i_0}, C_{i_1}, \ldots, C_{i_{g+1}}$ (for some g and some configurations $C_{i_0}, C_{i_1}, \ldots, C_{i_g}, C_{i_{g+1}}$). One can easily observe, that $\bar{\delta}_0^s \neq \tilde{\delta}_0^s$ and $\bar{\delta}_r^s \neq \tilde{\delta}_r^s$, since $\bar{w} \neq \tilde{w}$, $\bar{w}, \tilde{w} \in S_{r,s} \subseteq L_k^n \subseteq L_k$, and since $\bar{w}, \tilde{w}$ have the same (r,s) deletion $z_1z_2z_3$. Hence, $\hat{w} = z_1\bar{\delta}_0^s z_2\tilde{\delta}_r^s z_3 \notin L_k$.

We derive a contradiction by showing that A must accept $\hat{w}$. To do so we will prove that there is a computation of A on $\hat{w}$ from $C_{i_j}, C_{i_{j+1}}$ for every $j = 0, 1, 2, \ldots, g$. Let us consider the following two cases.

Case 1. No head scans $\bar{\delta}_0^s$ when A is in C_{i_j} on the input $\bar{w}$. Clearly, no head scans $\tilde{\delta}_0^s$ when A is in C_{i_j} on the input $\tilde{w}$, since $\bar{w}$ and $\tilde{w}$ are identical strings with exception for the corresponding substrings $\bar{\delta}_0^s \neq \tilde{\delta}_0^s$ and $\bar{\delta}_r^s \neq \tilde{\delta}_r^s$ of the same length n. Since $C_{i_1}, C_{i_2}, \ldots, C_{i_g}$ are *all* the configurations that are important for $\tilde{w}$ (recall that the sequence $C_{i_0}, C_{i_1}, \ldots, C_{i_{g+1}}$ is the profile of $\tilde{w}$, see above), and since *no* head scans $\tilde{\delta}_0^s$ when A is in C_{i_j} on the input $\tilde{w}$ (see above), then no head enters $\tilde{\delta}_0^s$, and hence, no head scans $\tilde{\delta}_0^s$ during the computation of A on $\tilde{w}$ from C_{i_j} to $C_{i_{j+1}}$ (with possible exception for the configuration $C_{i_{j+1}}$). It means that there is a computation of A on $\hat{w}$ from C_{i_j} to $C_{i_{j+1}}$ during which A behaves exactly as it does on $\tilde{w}$, since $\tilde{w}$ and $\hat{w}$ are identical strings with exception for the corresponding unscanned substrings $\tilde{\delta}_0^s$ and $\bar{\delta}_0^s$ of the same length n.

Case 2. No head scans $\bar{\delta}_r^s$ when A is in C_{i_j} on the input $\bar{w}$. The proof in this case is similar to the proof in the Case 1, but somewhat simpler, since we do not consider the string $\tilde{w}$ at all. Note that the roles of $\tilde{\delta}_0^s$ and $\bar{\delta}_0^s$ play $\bar{\delta}_r^s$ and $\tilde{\delta}_r^s$ in this case.

Since $\bar{\delta}_0^s$ and $\bar{\delta}_r^s$ are never scanned simultaneously during the computation of A on $\bar{w}$ (see the selection of r and s above), then the assumption of the Case 1 or the assumption of the Case 2 is satisfied for every $i = 0, 1, \ldots, g$. This completes the proof of the Theorem 3. □

Theorem 2, Theorem 3 and the inequality $\binom{\lfloor\sqrt{2}k\rfloor}{2} < k^2$ yield the following result.

Corollary 1. *Let $k \geq 1$. Then $\mathcal{L}((\lfloor(\sqrt{2}k\rfloor)$-DiDFA$) \subset \mathcal{L}((2k+2)$-DiDFA$)$.*

References

[1] Holzer, M.: Multi-head finite automata: data-independent versus data-dependent computations, *Theoretical Computer Science*, 286(2002), 97-116.

[2] Holzer, M.: Data-independent versus data-dependent computations on multi-head automata, *Doctoral Thesis, Universität Tübingen*, 1998.

[3] Holzer, M., Kutrib, M., Malcher, A.: Complexity of multi-head finite automata: Origins and directions, *Theoretical Computer Science*, 412(2011), 83-96.

[4] Monien, B.: Two-way multihead automata over a one-letter alphabet, *RAIRO Inform. Théor.*, 14(1980), 67-82.

[5] Yao, A. C., Rivest, R.: k+1 heads are better than k, *Journal of the ACM*, 25(2), (1978), 337-340.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de