On Stateless Multihead Finite Automata and Multihead Pushdown Automata

Pierluigi Frisco[a,*], Oscar H. Ibarra[b,1]

[a] *School of Mathematical and Computer Sciences, Heriot-Watt University, EH14 4AS Edinburgh, UK*
[b] *Department of Computer Science, University of California, Santa Barbara, CA 93106, USA*

Abstract

A stateless k-head two-way deterministic finite automaton (k-head 2DFA), has only one state, hence the designation stateless. Its transitions depends solely on the symbols currently scanned by its k heads, and in every such transition each head can move one cell left, right, or remain stationary. An input, which is delimited by end markers, is accepted if the machine, when started with all heads on the left end marker, reaches the configuration where all the heads are on the right end marker. The nondeterministic version is denoted by k-head 2NFA.

We prove that stateless $(k+1)$-head 2DFAs (resp., 2NFAs) are computationally more powerful than k-head 2DFAs (resp., 2NFAs), improving a recent result where it was shown that $(k+4)$ heads are better than k heads.

We also study stateless multihead pushdown automata in their two-way and one-way, deterministic and nonderministic variations and show that for all these varieties, $k+1$ heads allow more computational power than k heads. Finally, we give some characterizations of stateless multihead finite and multihead pushdown automata.

Key words: stateless multihead finite automata, stateless multihead pushdown automata, head hierarchies, characterizations

1. Introduction

Denote a two-way nondeterministic (deterministic) finite automaton by 2NFA (2DFA). Similarly denote the one-way variant by 1NFA (1DFA). We consider *stateless k-head 2NFAs* and define them as pairs (Σ, δ) where:

*Corresponding author
Email addresses: pier@macs.hw.ac.uk (Pierluigi Frisco), ibarra@cs.ucsb.edu (Oscar H. Ibarra)

[1] The work of Oscar H. Ibarra was supported in part by NSF Grant CCF-0524136 and a Distinguished Visiting Fellowship at the University of Liverpool from the UK Royal Academy of Engineering.

Σ is an *alphabet* with $\triangleright, \triangleleft \notin \Sigma$, $\triangleright$ is the left end marker and $\triangleleft$ is the right end marker;

δ is a finite *set of transitions* of the form $(a_1, \ldots, a_k) \to (d_1, \ldots, d_k)$, where $a_i \in \Sigma \cup \{\triangleright, \triangleleft\}$, $1 \leq i \leq k$, is the symbol read by the i-th head, while $d_i \in \{\mathsf{l}, \mathsf{s}, \mathsf{r}\}$ tells where the i-th head is to be moved (l, s and r, stand for *left*, *stay* and *right*, respectively).

If there is at most one transition for every combination of symbols $a_1, \ldots, a_k \in \Sigma \cup \{\triangleright, \triangleleft\}$, we refer to such an automaton as a *stateless k-head 2DFA*. If none of the transitions move any heads to the left, such an automaton is called a *stateless k-head 1NFA (1DFA)*.

For an input string $w \in \Sigma^*$, the machine works on a tape containing $\triangleright w \triangleleft$ and start with all heads reading the left end marker $\triangleright$. At every step of the computation, the symbols $a_1, \ldots, a_k$ currently scanned by all k heads are considered, any corresponding transition $a_1 \ldots a_k \to d_1 \ldots d_k \in \delta$ is chosen, and each i-th heads is moved according to d_i. The string is accepted if there exists a computation resulting in all k heads reading the end marker $\triangleleft$. As an example, the stateless 2-head 1DFA with transitions $(\triangleright, \triangleright) \to (\mathsf{s}, \mathsf{r})$, $(\triangleright, a) \to (\mathsf{s}, \mathsf{r})$, $(\triangleright, b) \to (\mathsf{r}, \mathsf{r})$, $(a, b) \to (\mathsf{r}, \mathsf{r})$, and $(b, \triangleleft) \to (\mathsf{r}, \mathsf{s})$ recognizes the language $L = \{a^n b^{n+1} \mid n \geqslant 0\}$.

Stateless multihead finite automata were first studied in [9] mainly with respect to the decision problems (e.g., emptiness). Later, hierarchies with respect to the number of heads were established in [5], where it was shown that stateless $(k+1)$-head 1DFAs (resp., 1NFAs) are computationally more powerful than stateless k-head 1DFAs (resp., 1NFAs), and stateless $(k+4)$-head 2DFAs (resp., 2NFAs) are computationally more powerful than stateless k-head 2DFAs (resp., 2NFAs). It was left open in [5], whether the "k+4" in the latter result can be improved. In this paper, we show that, indeed, stateless $(k+1)$-head 2DFAs (resp., 2NFAs) are computationally more powerful than stateless k-head 2DFAs (resp., 2NFAs).

We also look at stateless multihead pushdown automata in their two-way deterministic, two-way nondeterministic, one-way deterministic, and one-way nondeterministic versions (2DPDA, 2NPDA, 1DPDA, 1NPDA, respectively). The moves of these machines are similar to multihead automata, except that now, the transitions also depend on the stack. So the transition δ has form $(a_1, \ldots, a_k, Z) \to (d_1, \ldots d_k, \gamma)$, where Z is the symbol at the top of the pushdown stack, and $\gamma \in \Gamma^*$, with Γ the pushdown alphabet. (In this transition, Z is replaced by the string γ, with the rightmost symbol of γ becoming the new top of the stack. If γ is the null string, Z is popped from the stack.) We show that for all varieties of stateless multihead pushdown automata, $k+1$ heads are computationally more powerful than k heads.

Finally, we give some characterizations of stateless multihead finite and multihead pushdown automata.

2. Stateless Multihead Two-way Finite Automata

Our proofs involve "translations" (or reductions) to multihead automata with states. A *k-head 2NFA (2DFA) with states*, is a tuple $(\Sigma, \delta, S, s_0, s_f)$ where:

Σ is an *alphabet* with $\triangleright, \triangleleft \notin \Sigma$;

δ a finite *set of transitions* of the form $(s, a_1, \ldots, a_k) \to (s', d_1, \ldots, d_k)$ with $s, s' \in S$, being the *current* and *next* state, respectively, $a_1, \ldots, a_k \in \Sigma \cup \{\triangleright, \triangleleft\}$ being the symbols currently read by the k heads, and $d_1, \ldots, d_k \in \{\mathsf{l}, \mathsf{s}, \mathsf{r}\}^k$ being the movements of the k heads;

S is a finite *set of states*;

$s_0 \in S$ is the *initial state*;

$s_f \in S$ is the *final state*.

Such an automaton starts on a tape containing $\triangleright w \triangleleft$ in initial state s_0 with all heads reading $\triangleright$. If the automaton is in state s, then a transition of the form $(s, a_1, \ldots, a_k) \to (s', d_1, \ldots, d_k)$ can be applied only if head i reads a_i, $1 \leq i \leq k$. As a consequence of the application of such a transition, the state of the automaton changes into s' and head i moves according to d_i. The automaton accepts when it is in state s_f with all heads reading $\triangleleft$.

Lemma 2.1. *For $k \geq 2$, any* k*-head* 2DFA *with states can be simulated by a stateless* k*-head* 2DFA*.*

Proof. Let $M_1 = (\Sigma, \delta, S, s_0, s_f)$ be a k-head $2DFA$ with states such that $S = \{1, \ldots, |S|\}$, $s_0 = 1$ and $s_f = |S|$. In order to have a simpler proof we assume that the transitions of M_1 are of the form $(x, a_1, \ldots, a_k) \to (y, d_1, \ldots, d_k)$ where $d_1, d_2 \in \{\mathsf{l}, \mathsf{r}\}$ and $d_2, \ldots, d_k \in \{\mathsf{l}, \mathsf{r}, \mathsf{s}\}$. This assumption is not restrictive as given M_1, a $2DFA$ such that $d_1, d_2 \in \{\mathsf{l}, \mathsf{r}, \mathsf{s}\}$, then it is always possible to construct another $2DFA$ M_1' accepting the same language of M_1 and such that $d_1, d_2 \in \{\mathsf{l}, \mathsf{r}\}$. For each transition of M_1 for which head 1 or head 2 does not move, M_1' has two transitions: the corresponding head moves right and then back to the left if the head was not on $\triangleleft$; otherwise, it moves left and then right.

We define $M_2 = (\Sigma', \delta')$, a stateless k-head *2DFA* with

$$\Sigma' = \Sigma \cup \{q_{x,y}^{d_1,d_2}, {q'}_{x,y}^{d_1,d_2}, \bar{q}_{x,y}^{d_1,d_2}, \bar{q'}_{x,y}^{d_1,d_2} \mid (x, a_1, \ldots, a_k) \to (y, d_1, \ldots, d_k) \in \delta, \; d_1, d_2 \in \{\mathsf{l}, \mathsf{r}\}\}.$$

The elements in δ' are defined in the following.

Given a string $w = w_1 \ldots w_{|w|}$ over Σ we define the string $w' = p_1 p_2 w$ with:

$p_1 = w_1 t w_2 t \ldots w_{|w|} t$ where t is the lexicographic ordering of $tr_{x,y}$ for each $(x, a_1, \ldots, a_k) \to (y, d_1, \ldots, d_k) \in \delta$, and
$tr_{x,y} = \bar{q}_{x,y}^{\mathsf{l},\mathsf{l}} \; q_{x,y}^{\mathsf{l},\mathsf{l}} \; \bar{q}_{x,y}^{\mathsf{r},\mathsf{l}} \; q_{x,y}^{\mathsf{r},\mathsf{l}} \; q_{x,y} \; q_{x,y}^{\mathsf{l},\mathsf{r}} \; \bar{q}_{x,y}^{\mathsf{l},\mathsf{r}} \; q_{x,y}^{\mathsf{r},\mathsf{r}} \; \bar{q}_{x,y}^{\mathsf{r},\mathsf{r}}$;

$p_2 = w_1 t' w_2 t' \dots w_{|w|} t'$ where t' is the lexicographic ordering of $tr'_{x,y}$ for each $(x, a_1, \dots, a_k) \to (y, d_1, \dots, d_k) \in \delta$ and
$tr'_{x,y} = \bar{q'}^{\mathsf{r},\mathsf{l}}_{x,y}\ q'^{\mathsf{r},\mathsf{l}}_{x,y}\ \bar{q'}^{\mathsf{l},\mathsf{l}}_{x,y}\ q'^{\mathsf{l},\mathsf{l}}_{x,y}\ q'_{x,y}\ q'^{\mathsf{l},\mathsf{r}}_{x,y}\ \bar{q'}^{\mathsf{l},\mathsf{r}}_{x,y}\ q'^{\mathsf{r},\mathsf{r}}_{x,y}\ \bar{q'}^{\mathsf{r},\mathsf{r}}_{x,y}$.

With *lexicographic order* we mean that, for instance, $tr_{x,y}$ comes before $tr_{x',y'}$ if and only if $x < x'$ or $x = x'$ and $y \leq y$. During a computation of M_2 head 1 mainly reads p_1, head 2 mainly reads p_2, while the remaining heads mainly read w. The computations of M_2 can be logically divided in three parts: *initialization*, *simulation* and *termination*.

Initialization. In the initial configuration all the heads of M_2 read $\triangleright$ in w'. During initialization the heads move such that at the end of it:

head 1 reads $q_{s_0,y}$ in the $tr_{s_0,y}$ coming after a_1 in p_1;

head 2 reads $q'_{s_0,y}$ in the $tr'_{s_0,y}$ coming after a_1 in p_2;

the remaining heads read a_1 in w.

This occurs by the application of:
1: $\triangleright^k \to \mathsf{s}^2\mathsf{r}^{k-2}$;
2: $\triangleright^2\ \sigma^{k-2} \to \mathsf{sr}^{k-1}$ for $\sigma \in \Sigma$;
3: $\triangleright\ \sigma\ \varepsilon^{k-1} \to \mathsf{s}^2\mathsf{r}$ for $\sigma \in \Sigma$ and $\varepsilon \in \Sigma \cup \{\bar{q}^{t_1,t_2}_{x',y'},\ q^{t_1,t_2}_{x',y'},\ q_{x',y'} \mid (x', a_1, \dots, a_k) \to (y', d_1, \dots, d_k) \in \delta,\ t_1, t_2 \in \{\mathsf{l}, \mathsf{r}\}\}$;
4: $\triangleright\ \sigma\ \bar{q'}^{\mathsf{r},\mathsf{l}}_{s_0,y} \to \mathsf{sr}^{k-1}$ for $\sigma \in \Sigma$;
5: $\triangleright\ \bar{q}^{\mathsf{l},\mathsf{l}}_{s_0,y}\ \varepsilon^{k-2} \to \mathsf{s}^2\mathsf{r}^{k-2}$ for $\varepsilon \in \Sigma \cup \{\bar{q}^{t_1,t_2}_{x',y'},\ q'^{t_1,t_2}_{x',y'},\ q'_{x',y'} \mid (x', a_1, \dots, a_k) \to (y', d_1, \dots, d_k) \in \delta,\ t_1, t_2 \in \{\mathsf{l}, \mathsf{r}\}\} \setminus \{p_{2,|p_2|}\}$ where $p_{2,|p_2|}$ is the last symbol in p_2;
6: $\triangleright\ \bar{q}^{\mathsf{l},\mathsf{l}}_{s_0,y}\ p_{2,|p_2|} \to \mathsf{sr}^{k-1}$;
7: $\triangleright\ \varepsilon\ \sigma^{k-2} \to \mathsf{srs}^{k-2}$ for $\sigma \in \Sigma$ and $\varepsilon \in \Sigma \cup \{\bar{q}^{t_1,t_2}_{x',y'},\ q^{t_1,t_2}_{x',y'}, q_{x',y'} \mid (x', a_1, \dots, a_k) \to (y', d_1, \dots, d_k) \in \delta,\ t_1, t_2 \in \{\mathsf{l}, \mathsf{r}\}\} \cup \{\bar{q'}^{\mathsf{r},\mathsf{l}}_{s_0,y},\ q'^{\mathsf{r},\mathsf{l}}_{s_0,y},\ \bar{q'}^{\mathsf{l},\mathsf{l}}_{s_0,y},\ q'^{\mathsf{l},\mathsf{l}}_{s_0,y} \mid (s_0, a_1, \dots, a_k) \to (y, d_1, \dots, d_k) \in \delta\}$;
8: $\varepsilon\ q'_{s_0,y}\ \sigma^{k-2} \to \mathsf{rs}^{k-1}$ for $\sigma \in \Sigma$ and $\varepsilon \in \{w_1, \bar{q}^{\mathsf{l},\mathsf{l}}_{s_0,y},\ q^{\mathsf{l},\mathsf{l}}_{s_0,y},\ \bar{q}^{\mathsf{r},\mathsf{l}}_{s_0,y},\ q^{\mathsf{r},\mathsf{l}}_{s_0,y} \mid (s_0, a_1, \dots, a_k) \to (y, d_1, \dots, d_k) \in \delta\}$.

After the application of transition 1 the heads from 2 to k read the a_1 in p_1. Then, the application of transitions 2 let the heads from 3 to k to move to the right, the repeated application of transitions 3 let these heads to move to the right until they read the a_1 in p_2. The application of transitions 4 let heads from 2 to k to move to the right, then the repeated application of transitions 5 let heads from 3 to k to move to the right until they read the last symbol in p_2. The application of transition 6 let the head from 2 to k to move to the right, then the repeated application of transitions 7 let head 2 to move to the right until it reads $q'_{s_0,y}$. When this happens the repeated application of transitions 8 let head 1 to move to the right until it reads $q_{s_0,y}$.

Simulation. The simulation of transitions of M_1 starts and ends with head 1 and head 2 reading symbols of the kind $q_{x,y}$ and $q'_{x,y}$, respectively, for $x, y \in S$. This process is rather complex but it simply consists of the following. Let us assume that $p_1 = w_1 t_1 w_2 t_2 w_3 t_3 w_4$ (we know that $t_1 = t_2 = t_3 = t$ but here we need to refer to a specific t so we use subscripts to differentiate them) and that head 1 reads $q_{x,y}$ in t_2 for $(x, a_1, \ldots, a_k) \to (y, d_1, \ldots, d_k) \in \delta$. If $d_1 = \mathsf{r}$, then head 1 moves to the right until it reads $q_{y,z}$ in t_3, if instead $d_1 = \mathsf{l}$, then head 1 moves to the left until it reads $q_{y,z}$ in t_1. Head 2 moves in p_2 in a similar fashion depending on d_2.

It should be clear that this process requires more than one transition in M_2. The strings $tr_{x,y}$ and $tr'_{x,y}$ have substrings of pairs of symbols: one with a bar and one without. As we will see, the barred symbols in p_1 (p_2) are used to indicate that head 2 (head 1) passed to another t while scanning p_2 (p_1). The superscripts (pairs or elements in $\{\mathsf{l}, \mathsf{r}\}$) indicate in which direction head 1 and head 2 have to move.

In the following $\iota \in \{w_1, \ldots, w_{|w|}\}$ and $\sigma \in \Sigma$.

Phase 1. The transition is simulated on the heads from 3 until k, while head 1 moves to read $q^{d_1,d_2}_{x,y}$. Head 2 co-operates with head 1. This is performed by the following transitions divided into groups: (10, 11, 12), (13, 14), (15, 16) and (17, 18, 19). These groups perform similar operations.

9: $q_{x,y}\ q'_{x,y}\ \iota^{k-2} \to d_2 d_1 d_3 \ldots d_k$ for $(x, a_1, \ldots, a_k) \to (y, d_1, \ldots, d_k) \in \delta$;

10: $q^{\mathsf{l},\mathsf{r}}_{x,y}\ q'^{\mathsf{l},\mathsf{r}}_{x,y}\ \iota^{k-2} \to \mathsf{rs}^{k-1}$;

11: $\bar{q}^{\mathsf{l},\mathsf{r}}_{x,y}\ q'^{\mathsf{l},\mathsf{r}}_{x,y}\ \iota^{k-2} \to \mathsf{rs}^{k-1}$;

12: $q^{\mathsf{r},\mathsf{r}}_{x,y}\ \varepsilon\ \iota^{k-2} \to \mathsf{srs}^{k-2}$;

13: $q^{\mathsf{l},\mathsf{r}}_{x,y}\ q'^{\mathsf{l},\mathsf{l}}_{x,y}\ \iota^{k-2} \to \mathsf{srs}^{k-2}$;

14: $q^{\mathsf{l},\mathsf{r}}_{x,y}\ \varepsilon\ \iota^{k-2} \to \mathsf{srs}^{k-2}$;

15: $q^{\mathsf{r},\mathsf{l}}_{x,y}\ q'^{\mathsf{l},\mathsf{r}}_{x,y}\ \iota^{k-2} \to \mathsf{sls}^{k-2}$;

16: $q^{\mathsf{r},\mathsf{l}}_{x,y}\ \varepsilon\ \iota^{k-2} \to \mathsf{sls}^{k-2}$;

17: $q^{\mathsf{r},\mathsf{l}}_{x,y}\ q'^{\mathsf{l},\mathsf{l}}_{x,y}\ \iota^{k-2} \to \mathsf{ls}^{k-1}$;

18: $\bar{q}^{\mathsf{r},\mathsf{l}}_{x,y}\ q'^{\mathsf{l},\mathsf{l}}_{x,y}\ \iota^{k-2} \to \mathsf{ls}^{k-1}$;

19: $q^{\mathsf{l},\mathsf{l}}_{x,y}\ \varepsilon\ \iota^{k-2} \to \mathsf{sls}^{k-2}$;

for $\varepsilon \in \{q'^{t_1,t_2}_{x,y},\ \bar{q}'^{t_1,t_2}_{x,y},\ q'_{x,y} \mid (x, a_1, \ldots, a_k) \to (y, d_1, \ldots, d_k) \in \delta,$
$t_1, t_2 \in \{\mathsf{l}, \mathsf{r}\}\}$ and for $(x, a_1, \ldots, a_k) \to (y, d_1, \ldots, d_k) \in \delta$.

Transition 9 lets heads 3, $\ldots, k$ to move as similar heads in M_1 do. If $d_1 = d_2 = \mathsf{r}$, then head 1 has to read $q^{\mathsf{r},\mathsf{r}}_{x,y}$. This is accomplished by the application of transitions 10 and 11. Similar result is obtained by the application of transitions 13, 15, 17 and 18 for the remaining possible values of d_1 and d_2. Head 1 then reads $q^{d_1,d_2}_{x,y}$. Depending on what read by head 1, transitions 12, 14, 16 and 19 let head 2 to move until the next symbol in Σ.

Phase 2. Let us assume that head 1 reads $q_{x,y}^{d_1,d_2}$. When head 2 reads a symbol in w, head 1 moves to read $\bar{q}_{x,y}^{d_1,d_2}$, then head 2 moves in direction d_2 until it reads ${q'}_{y,z}^{d_1,g}$ (where $g = \mathsf{l}$ if $d_2 = \mathsf{r}$ and $g = \mathsf{r}$ if $d_2 = \mathsf{l}$). This is performed by the following transitions divided into groups: (20, 21), (22, 23), (24, 25) and (26, 27). These groups perform similar operations.

20: $q_{x,y}^{\mathsf{r},\mathsf{r}}\ \sigma\ \iota^{k-2} \to \mathsf{r}^2\mathsf{s}^{k-2}$;

21: $\bar{q}_{x,y}^{\mathsf{r},\mathsf{r}}\ \varepsilon\ \iota^{k-2} \to \mathsf{srs}^{k-2}$;

for $\varepsilon \in \{{q'}_{x',y'}^{t_1,t_2},\ {\bar{q}'}_{x'.y'}^{t_1,t_2},\ {q'}_{x',y'} \mid (x', a_1, \ldots, a_k) \to (y', d_1, \ldots, d_k) \in \delta, x' < y,$
$t_1, t_2 \in \{\mathsf{l}, \mathsf{r}\}\} \cup \{\bar{q}_{y,z}^{\mathsf{r},\mathsf{l}} \mid (y, a_1, \ldots, a_k) \to (z, d_1, \ldots, d_k) \in \delta\}$;

22: $q_{x,y}^{\mathsf{l},\mathsf{r}}\ \sigma\ \iota^{k-2} \to \mathsf{r}^2\mathsf{s}^{k-2}$;

23: $\bar{q}_{x,y}^{\mathsf{l},\mathsf{r}}\ \varepsilon\ \iota^{k-2} \to \mathsf{srs}^{k-2}$;

for $\varepsilon \in \{{q'}_{x',y'}^{t_1,t_2},\ {\bar{q}'}_{x'.y'}^{t_1,t_2},\ {q'}_{x',y'} \mid (x', a_1, \ldots, a_k) \to (y', d_1, \ldots, d_k) \in \delta, x' < y,$
$t_1, t_2 \in \{\mathsf{l}, \mathsf{r}\}\} \cup \{\bar{q}_{y,z}^{\mathsf{r},\mathsf{l}},\ {q'}_{y,z}^{\mathsf{r},\mathsf{l}},\ {\bar{q}'}_{y,z}^{\mathsf{l},\mathsf{l}} \mid (y, a_1, \ldots, a_k) \to (z, d_1, \ldots, d_k) \in \delta\}$;

24: $q_{x,y}^{\mathsf{r},\mathsf{l}}\sigma\ \iota^{k-2} \to \mathsf{l}^2\mathsf{s}^{k-2}$;

25: $\bar{q}_{x,y}^{\mathsf{r},\mathsf{l}}\ \varepsilon\ \iota^{k-2} \to \mathsf{sls}^{k-2}$;

for $\varepsilon \in \{{q'}_{x',y'}^{t_1,t_2},\ {\bar{q}'}_{x'.y'}^{t_1,t_2},\ {q'}_{x',y'} \mid (x', a_1, \ldots, a_k) \to (y', d_1, \ldots, d_k) \in \delta, x' > y,$
$t_1, t_2 \in \{\mathsf{l}, \mathsf{r}\}\} \cup \{\bar{q'}_{y,z}^{\mathsf{r},\mathsf{r}} \mid (y, a_1, \ldots, a_k) \to (z, d_1, \ldots, d_k) \in \delta\}$;

26: $q_{x,y}^{\mathsf{l},\mathsf{l}}\ \sigma\ \iota^{k-2} \to \mathsf{l}^2\mathsf{s}^{k-2}$;

27: $\bar{q}_{x,y}^{\mathsf{l},\mathsf{l}}\ \varepsilon\ \iota^{k-2} \to \mathsf{sls}^{k-2}$;

for $\varepsilon \in \{{q'}_{x',y'}^{t_1,t_2},\ {\bar{q}'}_{x'.y'}^{t_1,t_2},\ {q'}_{x',y'} \mid (x', a_1, \ldots, a_k) \to (y', d_1, \ldots, d_k) \in \delta, x' > y,$
$t_1, t_2 \in \{\mathsf{l}, \mathsf{r}\}\} \cup \{\bar{q'}_{y,z}^{\mathsf{r},\mathsf{r}},\ {q'}_{y,z}^{\mathsf{r},\mathsf{r}},\ {\bar{q}'}_{y,z}^{\mathsf{l},\mathsf{r}} \mid (y, a_1, \ldots, a_k) \to (z, d_1, \ldots, d_k) \in \delta\}$.

When head 2 reads, for instance, ${q'}_{y,z}^{d_1,g}$, then, because of the application of transition 20, head 1 moves as indicated by d_1 so to read $\bar{q}_{x,y}^{d_1,d_2}$. The repeated application of transition 21 let head 1 to move as indicated by d_1 until it reads a symbol in Σ. The lexicographic order plays an essential role: transitions 21, 23, 25 and 27 let head 2 to move until it reads the unique (because M_1 is deterministic) ${q'}_{y,z}^{d_1,g}$.

Phase 3. When head 2 reads ${q'}_{y,z}^{d_1,g}$ the transitions in the present phase let head 1 to move in direction d_1 until it reads a symbol in Σ. Because of the lexicographic order during phase 3 the transitions in phase 2 cannot be applied. This is performed by the following transitions divided into groups: (28, 29), (30, 31), (32, 33) and (34, 35). These groups perform similar operations.

28: $\bar{q}_{x,y}^{\mathsf{r},\mathsf{r}}\ {q'}_{y,z}^{\mathsf{r},\mathsf{l}}\ \iota^{k-2} \to \mathsf{rs}^{k-1}$;

29: $\varepsilon\ {q'}_{y,z}^{\mathsf{r},\mathsf{l}}\ \iota^{k-2} \to \mathsf{rs}^{k-1}$;

30: $\bar{q}_{x,y}^{\mathsf{l},\mathsf{r}}\ {q'}_{y,z}^{\mathsf{l},\mathsf{l}}\ \iota^{k-2} \to \mathsf{ls}^{k-1}$;

31: $\varepsilon{q'}_{y,z}^{\mathsf{l},\mathsf{l}}\ \iota^{k-2} \to \mathsf{ls}^{k-1}$;

32: $\bar{q}_{x,y}^{\mathsf{r},\mathsf{l}}\ {q'}_{y,z}^{\mathsf{r},\mathsf{r}}\ \iota^{k-2} \to \mathsf{rs}^{k-1}$;

33: $\varepsilon\ {q'}^{\mathsf{r},\mathsf{r}}_{y,z}\ \iota^{k-2} \to \mathsf{rs}^{k-1}$;

34: $\bar{q}^{\mathsf{l},\mathsf{l}}_{x,y}\ {q'}^{\mathsf{l},\mathsf{r}}_{y,z}\ \iota^{k-2} \to \mathsf{ls}^{k-1}$;

35: $\varepsilon\ {q'}^{\mathsf{l},\mathsf{r}}_{y,z}\ \iota^{k-2} \to \mathsf{ls}^{k-1}$;

for $\varepsilon \in \{q^{t_1,t_2}_{x',y'},\ \bar{q}^{t_1,t_2}_{x',y'},\ q^{t_1,t_2}_{x',y'} \mid (x', a_1, \ldots, a_k) \to (y', d_1, \ldots, d_k) \in \delta,\ t_1, t_2 \in \{\mathsf{l}, \mathsf{r}\}\}$.

Phase 4. Similarly to what performed by the transitions in phase 2, when head 1 reads a symbol in Σ and head 2 reads ${q'}^{d_1,g}_{y,z}$, the transitions in the present phase let head 2 to move to read $\bar{q'}^{d_1,g}_{y,z}$. Head 1 goes on moving in direction d_1 until it reads $q_{y,z}$. This is performed by the following transitions divided into groups: (36, 37), (38, 39), (40, 41) and (42, 43). These groups perform similar operations.

36: $\sigma {q'}^{\mathsf{r},\mathsf{l}}_{y,z}\ \iota^{k-2} \to \mathsf{rls}^{k-2}$;

37: $\varepsilon \bar{q'}^{\mathsf{r},\mathsf{l}}_{y,z}\ \iota^{k-2} \to \mathsf{rs}^{k-1}$;

for $\varepsilon \in \{q^{t_1,t_2}_{x',y'},\ \bar{q}^{t_1,t_2}_{x',y'},\ q_{x',y'} \mid (x', a_1, \ldots, a_k) \to (y', d_1, \ldots, d_k) \in \delta, x' < y,\ t_1, t_2 \in \{\mathsf{l}, \mathsf{r}\}\} \cup \{\bar{q}^{\mathsf{l},\mathsf{l}}_{y,z},\ q^{\mathsf{l},\mathsf{l}}_{y,z},\ \bar{q}^{\mathsf{r},\mathsf{l}}_{y,z},\ q^{\mathsf{r},\mathsf{l}}_{y,z}\}$;

38: $\sigma {q'}^{\mathsf{l},\mathsf{l}}_{y,z}\ \iota^{k-2} \to \mathsf{l}^2\mathsf{s}^{k-2}$;

39: $\varepsilon\ \bar{q'}^{\mathsf{l},\mathsf{l}}_{y,z} \iota^{k-2} \to \mathsf{ls}^{k-1}$;

for $\varepsilon \in \{q^{t_1,t_2}_{x',y'},\ \bar{q}^{t_1,t_2}_{x',y'},\ q_{x',y'} \mid (x', a_1, \ldots, a_k) \to (y', d_1, \ldots, d_k) \in \delta, x' > y,\ t_1, t_2 \in \{\mathsf{l}, \mathsf{r}\}\} \cup \{\bar{q}^{\mathsf{r},\mathsf{r}}_{y,z},\ q^{\mathsf{r},\mathsf{r}}_{y,z},\ \bar{q}^{\mathsf{l},\mathsf{r}}_{y,z},\ q^{\mathsf{l},\mathsf{r}}_{y,z}\}$;

40: $\sigma {q'}^{\mathsf{r},\mathsf{r}}_{y,z}\ \iota^{k-2} \to \mathsf{r}^2\mathsf{s}^{k-2}$;

41: $\varepsilon\ \bar{q'}^{\mathsf{r},\mathsf{r}}_{y,z}\ \iota^{k-2} \to \mathsf{rs}^{k-1}$;

for $\varepsilon \in \{q^{t_1,t_2}_{x',y'},\ \bar{q}^{t_1,t_2}_{x',y'},\ q_{x',y'} \mid (x', a_1, \ldots, a_k) \to (y', d_1, \ldots, d_k) \in \delta, x' < y,\ t_1, t_2 \in \{\mathsf{lr}\}\} \cup \{\bar{q}^{\mathsf{l},\mathsf{l}}_{y,z},\ q^{\mathsf{l},\mathsf{l}}_{y,z},\ \bar{q}^{\mathsf{r},\mathsf{l}}_{y,z},\ q^{\mathsf{r},\mathsf{l}}_{y,z}\}$;

42: $\sigma\ {q'}^{\mathsf{l},\mathsf{r}}_{y,z}\ \iota^{k-2} \to \mathsf{lrs}^{k-2}$;

43: $\varepsilon\ \bar{q'}^{\mathsf{l},\mathsf{r}}_{y,z}\ \iota^{k-2} \to \mathsf{rs}^{k-1}$;

for $\varepsilon \in \{q^{t_1,t_2}_{x',y'},\ \bar{q}^{t_1,t_2}_{x',y'},\ q_{x',y'} \mid (x', a_1, \ldots, a_k) \to (y', d_1, \ldots, d_k) \in \delta, x' > y,\ t_1, t_2 \in \{\mathsf{l}, \mathsf{r}\}\} \cup \{\bar{q}^{\mathsf{r},\mathsf{r}}_{y,z},\ q^{\mathsf{r},\mathsf{r}}_{y,z},\ \bar{q}^{\mathsf{l},\mathsf{r}}_{y,z},\ q^{\mathsf{l},\mathsf{r}}_{y,z}\}$.

Phase 5. When head 1 reads $q_{y,z}$, then head 2 moves to read $q'_{y,z}$. This is performed by the following transitions divided into groups: (44, 45), (46, 47), (48, 49) and (50, 51). These groups perform similar operations.

44: $q_{y,z}\ \bar{q'}^{\mathsf{r},\mathsf{l}}_{y,z}\ \iota^{k-2} \to \mathsf{srs}^{k-2}$;

45: $q_{y,z}\ \varepsilon\ \iota^{k-2} \to \mathsf{srs}^{k-2}$ for $\varepsilon \in \{{q'}^{\mathsf{r},\mathsf{l}}_{y,z},\ \bar{q'}^{\mathsf{l},\mathsf{l}}_{y,z},\ {q'}^{\mathsf{l},\mathsf{l}}_{y,z}\}$;

46: $q_{y,z}\ \bar{q'}^{\mathsf{l},\mathsf{l}}_{y,z}\ \iota^{k-2} \to \mathsf{srs}^{k-2}$;

47: $q_{y,z}\ {q'}^{\mathsf{l},\mathsf{l}}_{y,z}\ \iota^{k-2} \to \mathsf{srs}^{k-2}$;

48: $q_{y,z}\ \bar{q'}^{\mathsf{r},\mathsf{r}}_{y,z}\ \iota^{k-2} \to \mathsf{sls}^{k-2}$;

49: $q_{y,z}\ \varepsilon\ \iota^{k-2} \to \mathsf{srs}^{k-2}$ for $\varepsilon \in \{{q'}^{\mathsf{r},\mathsf{r}}_{y,z},\ \bar{q'}^{\mathsf{l},\mathsf{r}}_{y,z},\ {q'}^{\mathsf{l},\mathsf{r}}_{y,z}\}$;
50: $q_{y,z}\ \bar{q'}^{\mathsf{l},\mathsf{r}}_{y,z}\ \iota^{k-2} \to \mathsf{sls}^{k-2}$;
51: $q_{y,z}\ {q'}^{\mathsf{l},\mathsf{r}}_{y,z}\ \iota^{k-2} \to \mathsf{sls}^{k-2}$.

The simulation ends with head 1 and head 2 reading $q_{y,z}$ and $q'_{y,z}$, respectively.

Termination. When head 1 reads q_{x,s_f} and head 2 reads q'_{x,s_f}, then all the heads move to the right until they read $\triangleleft$. This happens by the application of the following transitions:
52: $q_{x,s_f}\ q'_{x,s_f}\ \sigma^{k-2} \to \mathsf{s}^2\mathsf{r}^{k-2}$ for $\sigma \in \Sigma$;
53: $\varepsilon^2\ \triangleleft^{k-2} \to \mathsf{r}^2\mathsf{s}^{k-2}$ for $\varepsilon \in \Sigma' \setminus \{\triangleleft\}$;
54: $\triangleleft^k \to \mathsf{s}^k$. ∎

We were not able to prove a result similar to Lemma 2.1 for $2NFA$ due to the following difficulty. In the proof of Lemma 2.1 the transitions in phase 2 let head 2 to 'search' for (the unique) ${q'}^{d_1,g}_{y,z}$ while head 1 reads either $q^{d_1,d_2}_{x,y}$ or $\bar{q}^{d_1,d_2}_{x,y}$. When this happens the transitions in phase 3 let head 1 to move while head 2 does not. If a $2NFA$ tries to do a similar thing, then, as there can be several $q^{d_1,d_2}_{x,y}$ and $\bar{q}^{d_1,d_2}_{x,y}$ for different y in p_1, then when head 2 reads ${q'}^{d_1,g}_{y,z}$ (for a certain y) and transitions in phase 3 are applied, then it can be that head 1 reads $q^{d_1,d_2}_{x,y'}$. In this case the transitions in phase 2 can be applied again leading not to a simulation of any transition in M_1.

We did not succeed either to avoid such behaviour or to let the computation of M_1 to halt in a non accepting configuration when such behaviour occurs.

If the number of heads of M_1 is at least 3, then a stateless $2NFA$ can simulate a $2NFA$ with states.

Lemma 2.2. *For $k \geq 3$, any* k*-head* 2NFA *with states can be simulated by a stateless* k*-head* 2NFA*.*

Proof. As the proof is very similar to the one of Lemma 2.1, we only sketch it here.

Let $M_1 = (\Sigma, \delta, S, s_0, s_f)$ be a k-head $2NFA$ with states such that $S = \{1, \ldots, |S|\}$, $s_0 = 1$ and $s_f = |S|$. As in the proof of Lemma 2.1 we assume that the transitions of M_1 are of the form $(x, a_1, \ldots, a_k) \to (y, d_1, \ldots, d_k)$ where $d_1, d_2 \in \{\mathsf{l}, \mathsf{r}\}$ and $d_2, \ldots, d_k \in \{\mathsf{l}, \mathsf{r}, \mathsf{s}\}$.

We define $M_2 = (\Sigma', \delta')$, a stateless k-head *2NFA* with:

$$\Sigma' = \{q^{d_1,d_2}_{x,y}, {q'}^{d_1,d_2}_{x,y}, \bar{q}^{d_1,d_2}_{x,y}, \bar{q'}^{d_1,d_2}_{x,y} \mid (x, a_1, \ldots, a_k) \to (y, d_1, \ldots, d_k) \in \delta, \\ d_1, d_2 \in \{\mathsf{l}, \mathsf{s}, \mathsf{r}\}\} \cup \Sigma \cup \{\bar{\sigma} \mid \sigma \in \Sigma\}.$$

The elements in δ' are defined in the following.

Given a string $w = w_1 \ldots w_{|w|}$ over Σ we define the string $w' = p_1 p_2 p_3 w$ with:

p_1 and p_2 defined as in the proof of Lemma 2.1;

$p_3 = w_1\bar{w}_1 \dots w_{|w|}\bar{w}_{|w|}$.

The vast majority of transitions are similar to the ones described in the proof of Lemma 2.1. The main differences are:

Phase 1. Head 3 moves two times in direction d_3. This can be performed by
$9'$: $q_{x,y}\ q'_{x,y}\ \iota^{k-2} \to \mathsf{s}^2 d_3 \mathsf{s}^{k-3}$;
$9''$: $q_{x,y}\ q'_{x,y}\ \bar{\sigma}\ \iota^{k-3} \to d_2 d_1 d_3 \dots d_k$;
for $(x, a_1, \dots, a_k) \to (y, d_1, \dots, d_k) \in \delta$ and $\sigma \in \Sigma$.

Phase 2. Head 2 can move on when reading a ${q'}^{d_1,g}_{y,z}$ until it reads ${q'}^{d_1,g}_{y',z}$ with $y' > y$. This is because there can be several with different z (as M_1 is nondeterministic). Anyhow, if head 2 reads ${q'}^{d_1,g}_{y',z}$ with $y' > y$, then M_2 halts in a non-accepting configuration (because the heads do not read $\triangleleft$). So, when head 2 reads a ${q'}^{d_1,g}_{y,z}$, then either it moves on or head 3 moves to the right to read a barred symbol and head 2 does no longer move in this phase.

Phase 3. Transitions 45, 47, 49 and 51 are such that head 3 moves to the left. In this way M_2 is ready to simulate another transition of M_1. ∎

We can now state the main results of this section:

Theorem 2.1. *For $k \geq 2$, stateless k-head* 2DFAs *are computationally more powerful than stateless $(k-1)$-head* 2DFAs.

Proof. The case $k = 2$ is obvious, since the language $\{a^n b^{n+1} \mid n \geq 0\}$ can be accepted by a stateless 2-head DFA but not by a stateless 1-head DFA, since 1-head DFAs (even with states) accept only regular sets.

Now let $k \geq 3$. Let L_1 be a language accepted a k-head 2DFA with states M_1 but not by any $(k-1)$-head 2DFA with states. Such a language exists [6]. Let M_2 be the stateless k-head 2DFA simulating M_1 as described in Lemma 2.1. Let $L_2 = L(M_2)$. We claim that L_2 cannot be accepted by a stateless $(k-1)$-head 2DFA. Suppose not, i.e., L_2 is accepted by a stateless $(k-1)$-head 2DFA M_3. We can then construct from M_3 a $(k-1)$-head 2DFA M_4 with states to accept the original language $L_1 = L(M_1)$ as follows.

When given $\triangleright w \triangleleft$, M_4 simulates the computation of M_3 on $\triangleright p_1 p_2 w \triangleleft$. But since M_4 is only given $\triangleright w \triangleleft$ it will use the finite-state control to keep track of the movements of each head when each head "moves" into the segment $p_1 p_2$. Clearly, since p_1 and p_2 are fixed patterns, this can be done.

We get a contradiction, since L_1 cannot be accepted by a $(k-1)$-head 2DFA with states. ∎

Similarly, since k-head 2NFAs with states are computationally more powerful than $(k-1)$-head 2NFAS (for $k \geq 2$) [6], we have the following result using Lemma 2.2. However, the case $k = 3$ is still open.

Theorem 2.2. *Stateless k-head* 2NFAs *are computationally more powerful than stateless $(k-1)$-head* 2NFAs *for $k = 2$ and $k \geq 4$.*

We note that in the proof of Lemma 2.1 $|\Sigma'| = |\Sigma| + 16|\delta|$, where $|\delta|$ is the number of transitions in δ, while $|w'| = 18|\delta||w| + |w|$ (as $|p_1| = |p_2| = 9|\delta||w|$). If extra heads are added, then the size of the alphabet Σ' and the length of string w' used by a stateless 2DFA to simulate a k-head 2DFA with states can substantially decrease.

Lemma 2.3. *Let $M_1 = (\Sigma, \delta, S, s_0, s_f)$ be a* k*-head* 2DFA *with states where* k≥ 1. *There is a stateless $(k+1)$-head 2DFA $M_2 = (\Sigma', \delta')$ simulating M_1 with $|\Sigma'| = |\Sigma| + 4|\delta|$. The machine M_2 can simulate M_1 with input $|w|$ using an input string of length $|w'| = 3|\delta| + |w||\delta| + |w|$.*

Proof. As in the proof of Lemma 2.1 we consider that the transitions are of the form $(x, a_1, \ldots, a_k) \rightarrow (y, d_1, \ldots, d_k) \in \delta$ are such that $d_1 \in \{\mathsf{l}, \mathsf{r}\}$.

We define $M_2 = (\Sigma', \delta')$, a stateless $(k+1)$-head *2DFA* with:

$\Sigma' = \Sigma \cup \{q^{\mathsf{l}}_{x,y}, q^{\mathsf{r}}_{x,y}, q'_{x,y}, q_{x,y} \mid (x, a_1, \ldots, a_k) \rightarrow (y, d_1, \ldots, d_k) \in \delta\}$.

The elements in δ' are defined in the following.

Given a string $w = w_1 \ldots w_{|w|}$ over Σ we define the string $w' = p_1 p_2 w$ with:

p_1 is the lexicographic ordering of $tr_{x,y} = q^{\mathsf{l}}_{x,y} q_{x,y} q^{\mathsf{r}}_{x,y}$ for $(x, a_1, \ldots, a_k) \rightarrow (y, d_1, \ldots, d_k) \in \delta$;

$p_2 = w_1 t' w_2 t' \ldots w_{|w|} t'$ where t' is the lexicographic order of $q'_{x,y}$ for $(x, a_1, \ldots, a_k) \rightarrow (y, d_1, \ldots, d_k) \in \delta$.

The heads from 1 to k of M_2 are associated with the heads from 1 to k, respectively, of M_1. The remaining head of M_2 is called *head-state*. During a computation of M_2 the head-state reads mainly p_1, head 1 reads mainly p_2, while the remaining heads read mainly w.

The computations of M_2 can be logically divided in three parts: *initialization*, *simulation* and *termination*. Initialization and termination are similar to the ones described in the proof of Lemma 2.1, and for this reason we omit them here.

Simulation. The simulation of transitions of M_1 starts and ends with head-state and head 1 reading symbols of the kind $q_{x,y}$ and $q'_{x,y}$, respectively, for $x, y \in S$ while the remaining heads read symbols in w. In such a configuration the transition $(x, a_1, \ldots, a_k) \rightarrow (y, d_1, \ldots, d_k) \in \delta$ is simulated by head 2, ..., head k of M_2. Moreover, if $d_1 = \mathsf{r}$ ($d_1 = \mathsf{l}$), then head-1 and head-state of M_2 move right (left). This occurs by the application of transition 1 in M_2.

When the head-state reads $q^{\mathsf{r}}_{x,y}$ head 1 of M_2 moves to the right until $\sigma \in \Sigma$ (if the head-state reads $q^{\mathsf{l}}_{x,y}$, then head 1 of M_2 moves to the left until $\sigma \in \Sigma$). This occurs by the application of transitions 2 ($2'$) in M_2.

When head-state reads $q^{\mathsf{r}}_{x,y}$ and head-1 of M_2 reads $\sigma \in \Sigma$, then head 1 moves to the right while head-state moves to the left (if head-state reads $q^{\mathsf{l}}_{x,y}$ and head-1 reads $\sigma \in \Sigma$, then head 1 and head-state move to the right). This occurs by the application of transitions 3 ($3'$).

Head 1 keeps then moving to the right until it reaches $q'_{y,z}$ (transition 4). When this happens the head-state moves (to the left if $y < x$, to the right if $y > x$) until it reads $q_{y,z}$ (transitions 5 and 5′). When this happens head-1 moves to the left until it reads an a (transition 6). When this happens M_2 is in a configuration similar to c.

The just indicated transitions are:

1: $q_{x,y}\ q'_{x,y}\ \iota^{k-1} \to d_1^2 d_2 \dots d_k$ for $(x, a_1, \dots, a_k) \to (y, d_1, \dots, d_k) \in \delta$;

2: $q^{\mathsf{r}}_{x,y}\ \varepsilon\ \iota^{k-1} \to \mathsf{srs}^{k-1}$
for $\varepsilon \in \{q'_{x',y'} \mid (x', a_1, \dots, a_k) \to (y', d_1, \dots, d_k) \in \delta\}$;

2′: $q^{\mathsf{l}}_{x,y}\ \varepsilon\ \iota^{k-1} \to \mathsf{srs}^{k-1}$
for $\varepsilon \in \{q'_{x',y'} \mid (x', a_1, \dots, a_k) \to (y', d_1, \dots, d_k) \in \delta\}$;

3: $q^{\mathsf{r}}_{x,y}\ \sigma\ \iota^{k-1} \to \mathsf{lrs}^{k-1}$ for $\sigma \in \Sigma$;

3′ $q^{\mathsf{l}}_{x,y}\ \sigma\ \iota^{k-1} \to \mathsf{r}^2\mathsf{s}^{k-1}$ for $\sigma \in \Sigma$;

4: $q_{x,y}\ q'_{x',y'}\ \iota^{k-1} \to \mathsf{srs}^{k-1}$ for $x' < y$;

5: $q_{x,y}\ q'_{y',z'}\ \iota^{k-1} \to \mathsf{rs}^k$ for $y' > x$;

5′: $q_{x,y}\ q'_{y',z'}\ \iota^{k-1} \to \mathsf{ls}^k$ for $y' < x$;

6: $q_{y,z}\ \varepsilon\ \iota^{k-1} \to \mathsf{sls}^{k-1}$
for $\varepsilon \in \{q'_{y',z'} \mid (y', a_1, \dots, a_k) \to (z', d_1, \dots, d_k) \in \delta,\ y' \le y\}$.

The result follows. ∎

Lemma 2.4. *Let $M_1 = (\Sigma, \delta, S, s_0, s_f)$ be a* k*-head* 2DFA *with states where* k≥ 1. *There is a stateless $(k+2)$-head 2DFA $M_2 = (\Sigma', \delta')$ simulating M_1 with $|\Sigma'| = |\Sigma| + 4|\delta|$. The system M_2 can simulate M_1 with input $|w|$ using an input string of length $|w'| = 4|\delta| + |w|$.*

Proof. We define $M_2 = (\Sigma', \delta')$, a stateless $(k+1)$-head *2DFA* with:

$\Sigma' = \Sigma \cup \{q_{x,y}, f_{x,y}, q'_{x,y}, q''_{x,y} \mid (x, a_1, \dots, a_k) \to (y, d_1, \dots, d_k) \in \delta\}$.

The elements in δ' are defined in the following.

Given a string $w = w_1 \dots w_{|w|}$ over Σ we define the string $w' = p_1 p_2 w$ with:

p_1 is the lexicographic ordering of $f_{x,y} q_{x,y}$ for $(x, a_1, \dots, a_k) \to (y, d_1, \dots, d_k) \in \delta$;

p_2 is the lexicographic ordering of $q'_{x,y}$ followed by the lexicographic ordering of $q''_{x,y}$ for $(x, a_1, \dots, a_k) \to (y, d_1, \dots, d_k) \in \delta$.

The heads from 1 to k of M_2 are associated with the heads from 1 to k, respectively, of M_1. The remaining 2 heads of M_2 are called *head-state 1* and *head-state 2*. During a computation of M_2 the head-state 1 reads mainly p_1, head-state 2 reads mainly p_2, while while the remaining heads read mainly w.

The computations of M_2 can be logically divided in three parts: *initialization*, *simulation* and *termination*. Initialization and termination are similar to the ones described in the proof of Lemma 2.1, and for this reason we omit them here.

Simulation. The simulation of transitions of M_1 starts and ends with head-state 1 reading a symbol of the kind $q_{x,y}$ and head-state 2 reading a symbol of the kind $q'_{x,y}$ for $x, y \in S$ while the remaining heads read symbols in w. In such a configuration the transition $(x, a_1, \ldots, a_k) \rightarrow (y, d_1, \ldots, d_k) \in \delta$ is simulated by head 1, ..., head k of M_2. Moreover, head-state 1 moves to the left while head-state 2 moves to the right (transition 1).

When head-state 1 reads $f_{x,y}$, then head-state 2 moves to the right until it reads $q''_{y,z}$ (transitions 2). When this happens head-state 1 moves to the right or to the left (depending on the lexicographic order) until it reads $q_{y,z}$ (transitions 3 and 3′). When this happens head-state 2 moves to the right or to the left (depending on the lexicographic order) until it reads $q'_{y,z}$ (transitions 5).

The just indicated transitions are:

1: $q_{x,y}\ q'_{x,y}\ \iota^k \rightarrow \mathsf{lr}d_1 \ldots d_k$ for $(x, a_1, \ldots, a_k) \rightarrow (y, d_1, \ldots, d_k) \in \delta$;

2: $f_{x,y}\ \varepsilon\ \iota^k \rightarrow \mathsf{srs}^k$ for $\varepsilon \in \{q'_{x',y'} \mid (x', a_1, \ldots, a_k) \rightarrow (y', d_1, \ldots, d_k) \in \delta\} \cup$
$\{q''_{x'',y''} \mid (x'', a_1, \ldots, a_k) \rightarrow (y'', d_1, \ldots, d_k) \in \delta,\ x'' < y\}$;

3: $f_{x',y'}\ q''_{y,z}\ \iota^k \rightarrow \mathsf{rs}^{k+1}$ for $y \geq x$;

3′: $f_{x',y'}\ q''_{x,y}\ \iota^k \rightarrow \mathsf{ls}^{k+1}$ for $y < x$;

4: $q_{x',y'}\ q''_{y,z}\ \iota^k \rightarrow \mathsf{rs}^{k+1}$ for $y > x$;

4′: $q_{x',y'}\ q''_{y,z}\ \iota^k \rightarrow \mathsf{ls}^{k+1}$ for $y < x$;

5: $q_{y,z}\ \varepsilon\ \iota^k \rightarrow \mathsf{sls}^k$ for $\varepsilon \in \{q''_{x',y'} \mid (x', a_1, \ldots, a_k) \rightarrow (y', d_1, \ldots, d_k) \in \delta,$
$x' \leq y\} \cup \{q'_{x'',y''} \mid (x'', a_1, \ldots, a_k) \rightarrow (y'', d_1, \ldots, d_k) \in \delta,\ x'' > y\}$.

The result follows. ∎

Finally, when the number of heads of the simulating stateless 2DFA is $k+3$, we have [5, Lemma 1]:

Lemma 2.5. *Let $M_1 = (\Sigma, \delta, S, s_0, s_f)$ be a* k*-head* 2DFA *with states where* k≥ 1. *There is a stateless $(k+3)$-head 2DFA $M_2 = (\Sigma', \delta')$ simulating M_1 with $|\Sigma'| = |\Sigma| + |\delta|$. The system M_2 can simulate M_1 with input $|w|$ using an input string of length $|w'| = |\delta| + |w|$.*

We do not know if the cardinality of Σ' and the length of w' in Lemmas 2.1, 2.3, 2.4 and 2.5 are optimal. Here we simply wanted to point out how the cardinality and length decrease when we increase the number of heads.

3. Stateless Multihead Pushdown Automata

Let M be a k-head 2DPDA (two-way deterministic pushdown automaton) over input alphabet Σ. An input to M is a tape of the form $\triangleright w \triangleleft$, where $w \in \Sigma^*$ with $\triangleright$ and $\triangleleft$ the left and right end markers. Let $\triangleright w \triangleleft = a_1 \ldots a_n$. (Thus, $n \geq 2$, and $a_1 = \triangleright$ and $a_n = \triangleleft$.) Let $\{1, \ldots, m\}$ be the sets of states. Initially, the stack contains B (the bottom of the stack) and all k heads are on a_1 (i.e., on $\triangleright$). The transitions of M are of the form $(i, x_1, \ldots, x_k, z) \rightarrow (j, d_1, \ldots, d_k, \gamma)$. The transition means that when M is in state i, the heads $H_1, \ldots, H_k$ read $x_1, \ldots, x_k \in \Sigma \cup \{\triangleright, \triangleleft\}$, and the *top of the stack* (TOS) is z, then M changes state to j, moves head H_s in direction $d_s \in \{\mathsf{l}, \mathsf{s}, \mathsf{r}\}$, and replaces z with the

string γ. The tape $w = a_1 \ldots a_n$ (actually, $a_2 \ldots a_{n-1}$, without the end markers) is accepted if and only if M eventually reaches the configuration where all the heads are on a_n (i. e., on $\triangleleft$) and the stack is empty (there is no move on empty stack). We assume that no head falls off the input tape.

We describe a stateless k-head 2DPDA M' simulating M. The input alphabet of M' is $\Sigma' = \Sigma \cup \{\triangleright, \triangleleft, \#\} \cup \{i^{\mathsf{l}}, i^{\mathsf{r}} \mid 1 \leq i \leq m\}$. The left and right end markers of inputs to M' are now $\triangleright'$ and $\triangleleft'$, respectively. Note that $\triangleright, \triangleleft$ are now considered input symbols in Σ'

Let $p = 1^{\mathsf{l}}2^{\mathsf{l}} \ldots m^{\mathsf{l}}1^{\mathsf{r}}2^{\mathsf{r}} \ldots m^{\mathsf{r}}$. For an input $w = a_1 \ldots a_n$ to M, let $w' = \triangleright' a_1 p a_2 p \ldots a_n p \# a_1 \ldots a_n \triangleleft'$. We refer to w' as the encoding of w. We construct M' such that M accepts w if and only if M' accepts w'. For inputs to M' that are not valid encodings of inputs to M, we do not care. We now briefly describe the operation of M' when given input $w' = \triangleright' a_1 p a_2 p \ldots a_n p \# a_1 \ldots a_n \triangleleft'$. For convenience, we also call the heads of M', $H_1, \ldots, H_k$.

M' simulates the computation of M on $w = a_1 \ldots a_n$ on input $w' = \triangleright' a_1 p a_2 p \ldots a_n p \# a_1 \ldots a_n \triangleleft'$ as follows.

1. While H_1 remains on $\triangleright'$, heads $H_2, \ldots, H_k$ are moved to $\#$.
2. Then $H_1, \ldots, H_k$ are moved right one cell. So now, H_1 is on the first a_1 and $H_2, \ldots, H_k$ are on the second a_1 (directly to the right of $\#$).

   In what follows, $H_2, \ldots, H_k$ will simulate the corresponding heads of M on $a_1 \ldots a_n$. H_1 will simulate H_1 of M on the segment $a_1 p a_2 p \ldots a_n p$. For convenience, let $w_1 = a_1 p a_2 p \ldots a_n p$ and $w_2 = a_1 \ldots a_n$.
3. The current state i of M is stored on the TOS of M' along with the stack symbol, as a pair (z, i).
4. At the beginning of each simulation step, H_1 will be on some symbol a_t of w_1 and $H_2, \ldots, H_k$ are on w_2, and the TOS is some pair (z, i). (Initially, the stack contains only $(B, 1)$, where 1 is the initial state of M.)
5. If in the move, M' is supposed to replace (z, i) by $(z_1 \ldots z_r, j)$, $r > 0$, M' moves $H_2, \ldots, H_k$ as in M and, depending on whether H_1 is supposed to move right, left, or remain stationary of a_t, M' does the following:
   (a) moves H_1 right and replaces the TOS (z, i) by $(z_1 \ldots (z_r, j^{\mathsf{r}})$;
   (b) moves H_1 left and replaces the TOS (z, i) by $(z_1 \ldots (z_r, j^{\mathsf{l}})$;
   (c) does not move H_1 but replaces the TOS (z, i) by $(z_1 \ldots (z_r, j)$.

   Then, we consider three cases.

   Case 1: TOS is (z_r, j^{r}). Then H_1 moves right until it reads a_{t+1}. It then replaces the TOS (z_r, j^{r}) by (z_r, j), and continues the simulation of the next step of M;

   Case 2: TOS is (z_r, j^{l}). Then H_1 moves left until it reads a_{t-1}. It then replaces the TOS (z_r, j^{l}) by (z_r, j), and it continues the simulation of the next step of M;

   Case 3: TOS is (z_r, j). M' continues the simulation of the next step of M.

6. If in the move, M' is supposed to replace (z, i) by (ε, j) (i.e., the stack is popped), then M' moves $H_2, \ldots, H_k$ as in M and, depending on whether H_1 is supposed to move right, left, or remain stationary of a_t does the following:
   (a) Moves H_1 right and replaces the TOS (z, i) by $(\varepsilon, j^{\mathsf{r}})$;
   (b) Moves H_1 left and replaces the TOS (z, i) by $(\varepsilon, j^{\mathsf{l}})$;
   (c) Moves H_1 right and replaces the TOS (z, i) by $(\varepsilon, j^{\mathsf{s}})$.

   Again, we consider three cases:

   Case 4: TOS is $(\varepsilon, j^{\mathsf{r}})$. Then H_1 moves right until it reads j^{r}. Then it pops the stack. Let the new top be y. Then M' replaces y by (y, j^{r}). Then M' operates like Case 1;

   Case 5: TOS is $(\varepsilon, j^{\mathsf{l}})$. Then H_1 moves left until it reads j^{l}. Then it pops the stack. Let the new top be y. Then M' replaces y by (y, j^{l}). Then M' operates like Case 2;

   Case 6: TOS is $(\varepsilon, j^{\mathsf{s}})$. Then H_1 moves right until it reads j^{r}. Then it pops the stack. Let the new top be y. Then M' replaces y by (y, j^{l}). Then M' operates like Case 2.

It follows that M' can simulate M provided the input to M' is well-formed as described above.

Clearly, the above construction also works for nondeterministic machines, i.e., 2NPDAs.

Theorem 3.1. *For $k \geq 2$, stateless k-head 2DPDAs are computationally more powerful than stateless $(k-1)$-head 2DPDAs.*

Proof. Let L be a language accepted a k-head 2DPDA with states M but not by any $(k-1)$-head 2DPDA with states. Such a language exists [3]. Let M' be the stateless k-head 2DPDA simulating M as described above. Let $L' = L(M')$. We claim that L cannot be accepted by a stateless $(k-1)$-head 2DPDA. Suppose not, i.e., L' is accepted by a stateless $(k-1)$-head 2DPDA M''. We can then construct from M'' a $(k-1)$-head 2DPDA M''' with states to accept the original language $L = L(M)$ as follows.

When given $a_1 \ldots a_n$, M''' simulates the computation of M'' on $\triangleright' a_1 p \ldots a_n p \# a_1 \ldots a_n \triangleleft'$. But since M''' is only given $a_1 \ldots a_n$ it will use the finite-state control to keep track of the movements of each head when each head "moves" into the segment $\triangleright a_1 p a_2 p \ldots a_n p \#$. Clearly, since p is a fixed pattern, this can be done. We omit the details. ∎

Similarly, since k-head 2NPDAs with states are computationally more powerful than those with k-heads [3], we have:

Theorem 3.2. *For $k \geq 2$, stateless k-head 2NPDAs are computationally more powerful than stateless $(k-1)$-head 2NPDAs.*

Turning now to the one-way varieties, consider the following language over the alphabet $\{a, b, \#, @\}$:

$$L_k = \{u_{\frac{k(k-1)}{2}} \# u_{\frac{k(k-1)}{2}-1} \# \ldots \# u_2 \# u_1 @ v_1 \# v_2 \# \ldots \# v_{\frac{k(k-1)}{2}-1} \# v_{\frac{k(k-1)}{2}} \mid u_i, v_i \in \{a,b\}^*, u_i = v_i\}.$$

It was shown in [10] that L_k can be accepted by a k-head 1DFA with states, but cannot be accepted by any $(k-1)$-head 1NFA with states. Later in [1] it was shown that, in fact, L_k cannot be accepted by any $(k-1)$-head 1NPDA with states.

Recently, in [5], it was shown that a language L'_k, which is a "padded" version of the language L_k, has the following properties:

1. L'_k can be accepted by a stateless k-head 1DFA, and therefore also by a stateless k-head 1DPDA;
2. If L'_k can be accepted by a stateless $(k-1)$-head 1NFA, then the unpadded version L_k can be accepted by a $(k-1)$-head 1NFA with states, and therefore also by a $(k-1)$-head 1NPDA with states.

In fact, following the argument in item 2 described in [5]), L'_k cannot be accepted by $(k-1)$-head 1NPDA with states.

Hence, L'_k can be accepted by a stateless k-head 1DFA (hence also by a stateless k-head 1DPDA) but not by any stateless $(k-1)$-head 1NPDA.

Theorem 3.3. *There are languages accepted by stateless k-head 1DFAs that cannot be accepted by stateless $(k-1)$-head 1NPDAs.*

Corollary 3.1. *For $k \geq 2$, stateless k-head 1DPDAs (resp., 1NPDAs) are computationally more powerful than stateless $(k-1)$-head 1DPDAs (resp., 1NPDAs).*

4. Characterizations

Clearly, any multihead 2DFA (resp., 2NFA, 2DPDA, 2NPDA) with n states can be simulated by a corresponding stateless machine by adding $log_2 n$ extra heads to keep track of the states. Each head is positioned on the left end marker (to represent 0) and the cell to its right (to represent 1). Thus, $log_2 n$ heads can keep track of n states.

Theorem 4.1. *Let L be a language. Then:*

1. *L is accepted by a stateless multihead 2NPDA;*
2. *L is accepted by a stateless multihead 2DPDA;*
3. *L is accepted by a multihead 2NPDA with states;*
4. *L is accepted by a multihead 2DPDA with states;*
5. *L is accepted by a deterministic TM in n^c time for some constant c.*

The above follows from [2], where the equivalence of items 3, 4, and 5 was shown.

In fact, for stateless multihead nondeterministic pushdown automata, the following more precise characterization was recently shown in [4]:

Theorem 4.2. *A language L is accepted by a stateless k-head 2NPDA (resp., 1NPDA) if and only if it is accepted by a k-head 2NPDA (resp., 1NPDA) with states.*

One can also easily show the following:

Theorem 4.3. *A language L can be accepted by stateless multihead 1DPDA (resp., 1NPDA, 1DFA, 1NFA) if and only if it can be accepted by a multihead 1DPDA (resp., 1NPDA, 1DFA, 1NFA) with states M with the following property: when M enters a state, say q, it can remain in that state and move the heads (and change the stack) but once it leaves q and it enters another state, say q', then M cannot reenter state q.*

It is well known that multihead 2DFAs (resp., 2NFAs) are equivalent to *log n* space-bounded deterministic (nondeterministic) Turing machines. Hence, we have:

Theorem 4.4. *Stateless multihead 2DFAs (resp., 2NFAs) are equivalent to log n space-bounded deterministic (resp., nondeterministic) Turing machines.*

It is a long-standing open problem whether *log n* space-bounded deterministic Turing machines are equivalent to *log n* space-bounded nondeterministic Turing machines. Here we show:

Theorem 4.5. *If every language accepted by a stateless 3-head 2NFA can be accepted by a multihead 2DFA with states, then log n space-bounded deterministic Turing machines are equivalent to log n space-bounded nondeterministic Turing machines.*

Proof. We will use the fact that this theorem is true when the 3-head 2NFA has states [7]. (In fact, the result in [7] is already true for 2 heads.)

Let L_1 be a language accepted by a 3-head 2NFA. Then, the language L_2 (defined in the proof of Lemma 2.2) can be accepted by a stateless 3-head 2NFA. If L_2 can be accepted by a multihead 2DFA with states, then we can easily construct a multihead 2DFA with states accepting L_1.

The result follows. ∎

It would be interesting to know if the 3-head above can be reduced to 2-head.

5. Conclusion

We showed that stateless $(k+1)$-head 2DFA (resp., 2NFAs) are computationally more powerful than k-head 2DFAs (resp., 2NFAS) for $k \geq 1$ (resp., $k = 1$ and $k \geq 3$), improving recent results in [5]. We also proved similar results for stateless multihead pushdown automata. Some interesting problems remain.

For example, we conjecture that stateless 3-head 2NFAs are computationally more powerful than stateless 2-head 2NFAs, but we have no proof at this time.

We do not know if the relations between the number of heads, the size of Σ' and the length of w' presented in Section 2 are optimal. We think that such study, relating some features of the system to the size of the alphabet used in the simulation and the length of the input string, is important and that it is worth pursuing for other computing devices as well. We believe that the precise relationships between these measures depend on the kind of data structure (strings, multiset, sets, etc.) on which a system operates and on the 'power' of the transitions (rules, operation, etc.) used by the system.

References

[1] M. Chrobak and M. Li. k+1 heads are better than k for PDA's. *Proceedings of the 27th Annual Symp. on Foundations of Computer Science*, 361-367, 1986.

[2] S. Cook. Variations on pushdown machines. *Proceedings of the Annual ACM Symp. on Theory of Computing*, 229-231, 1969.

[3] O. H. Ibarra. On two-way multihead automata. *J. of Computer and System Sciences*, 7: 28–36, 1973.

[4] O. H. Ibarra and I. Potapov. On restricted multipushdown automata: properties and decision problems. In preparation.

[5] O. H. Ibarra, J. Karhumaki, and A. Okhotin. On stateless multihead automata: hierarchies and the emptiness problem. *Proceedings of LATIN 2008: Theoretical Informatics*, Lecture Notes in Computer Science, 94-105, 2008.

[6] B. Monien. Two-way multihead automata over a one-letter alphabet. *RAIRO Informatique theoretique*, 14(1): 67–82, 1980.

[7] I. H. Sudborough. On tape-bounded complexity classes and multi-head finite automata *Proc. of 14th Annual Symp. on Foundations of Computer Science*, 138-144, 1973.

[8] A. L. Rosenberg. On multi-head finite automata. *IBM Journal of Research and Development*, 10:5, 388-394, 1966.

[9] L. Yang, Z. Dang, and O. H. Ibarra. On stateless automata and P systems. *Pre-Proceedings of Workshop on Automata for Cellular and Molecular Computing*, August 2007.

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