Path-Bounded Four-Dimensional Finite Automata

YASUO UCHIDA and TAKAO ITO
Ube National College of Technology
Department of Business Administration
2-14-1,Tokiwadai,Ube City
Yamaguchi 755-8555, JAPAN
uchida@ube-k.ac.jp

MAKOTO SAKAMOTO,RYOJU KATAMUNE,KAZUYUKI UCHIDA,
HIROSHI FURUTANI,MICHIKO KONO, and SATOSHI IKEDA
Department of Computer Science and Systems Engineering
University of Miyazaki
1-1,Gakuen Kibanadai Nishi,Miyazaki City
Miyazaki 889-2192, JAPAN
sakamoto@cs.miyazaki-u.ac.jp

TSUNEHITO YOSHINAGA
Department of Computer Science and Electronic Engineering
Tokuyama College of Technology
Gakuendai,Shunan,Yamaguchi
Yamaguchi 745-8585, JAPAN
yoshinaga@tokuyama.ac.jp

Abstract: The comparative study of the computational powers of deterministic and nondeterministic computations is one of the central tasks of complexity theory. This paper investigates the computational power of nondeterministic computing devices with restricted nondeterminism. There are only few results measuring the computational power of restricted nondeterminism. In general, there are three possibilities to measure the amount of nondeterminism in computation. In this paper, we consider the possibility to count the number of different nondeterministic computation paths on any input. In particular, we deal with seven-way four-dimensional finite automata with multiple input heads operating on four-dimensional input tapes.

Key–Words: computational complexity, finite automaton, four-dimension, multihead, path-bounded

1 Introduction

The question of whether processing four-dimensional digital patterns is much difficult than three-dimensional ones is of great interest from the theoretical and practical standpoints. In recent years, due to the advances in many application areas such as moving image processing, computer animation, and so on, the study of four-dimensional pattern processing has been of crucial importance. Thus, the study of four-dimensional automata as the computational model of four-dimensional pattern processing has been meaningful. For example, in [8,9], a four-dimensional finite automaton was proposed as a natural extension of the three-dimensional finite automaton to four dimensions. On the other hand, the comparative study of the computational powers of deterministic computations is one of the central tasks of complexity theory.

In this paper, we investigate the computational power of nondeterministic computing devices with restricted nondeterminism. However, there are only few results [1-4] measuring the computational power of restricted nondeterminism. In general, there are three possibilities to measure the amount of nondeterminism in computation. One possibility is to count the number of advice bits (nondeterministic guesses) in particular nondeterministic computations, and the second possibility is to count the number of accepting computation paths. The third possibility is to count the number of different nondeterministic computation paths on any input. This paper considers the third one. In particular, the paper investigates a hierarchy on the degree of nondeterminism of seven-way four-dimensional (simple) multi-head finite automata as a natural extension of the five-way three-dimensional (simple) multi-head finite automata [5]. Furthermore, we investigate a relationship between the accepting powers of nondeterminism and self-verifying nondeterminism for seven-way four-dimensional (simple) multihead finite automata with the number of computation paths restricted (see Figure 1).

2 Preliminaries

Let $\Sigma$ be a finite set of symbols. A four-dimensional tape over $\Sigma$ is a four-dimensional rectangular array
of elements of $\sum$. The set of all four-dimensional tapes over $\sum$ is denoted by $\sum^4$. Given a tape $x \in \sum^4$, for each integer $j$ ($1 \leq j \leq 4$), we let $l_j(x)$ be the length of $x$ along the $j$th axis. The set of all $x \in \sum^4$ with $l_1(x) = n_1, l_2(x) = n_2, l_3(x) = n_3$, and $l_4(x) = n_4$ is denoted by $\sum_{(n_1,n_2,n_3,n_4)}$. When $1 \leq i_j \leq l_j(x)$ for each $j(1 \leq j \leq 4)$, let $x(i_1,i_2,i_3,i_4)$ denote the symbol in $x$ with coordinates $(i_1,i_2,i_3,i_4)$. Furthermore, we define

$$x[(i_1,i_2,i_3,i_4),(i_1',i_2',i_3',i_4')],$$

when $1 \leq i_j \leq l_j(x)$ for each integer $j(1 \leq j \leq 4)$, as the four-dimensional input tape $y$ satisfying the following conditions:

(i) for each $j(1 \leq j \leq 4)$, $l_j(y) = l_j(x) + 1$;

(ii) for each $r_1, r_2, r_3, r_4$ ($1 \leq r_1 \leq l_1(y)$, $1 \leq r_2 \leq l_2(y)$, $1 \leq r_3 \leq l_3(y)$, $1 \leq r_4 \leq l_4(y)$), $y(r_1,r_2,r_3,r_4) = x(r_1+1,r_2+1,r_3+1,r_4+1)$.

For each $(i_1,i_2,i_3,i_4)$, $(i_1',i_2',i_3',i_4')$, we use the symbol $x[(i_1,i_2,i_3,i_4),(i_1',i_2',i_3',i_4')]$ the $[(i_1,i_2,i_3,i_4),(i_1',i_2',i_3',i_4')]$-segment of $x$.

A seven-way four-dimensional multidhead finite automaton (SV4-MHFA) \cite{5} is a finite automaton with multiple input heads operating on four-dimensional input tapes surrounded by boundary symbols #$'$s. These heads can move east, west, south, north, up, down, in the future, but not in the past. A seven-way four-dimensional simple multidhead finite automaton (SV4-SMHFA) is an SV4-MHFA which has only one reading head and other counting heads which can only detect whether they are on the boundary symbols or a symbol in the input alphabet.

When a four-dimensional input tape $x$ is presented to a four-dimensional device $M$, $M$ starts in its initial state with all its heads on $x(1,1,1,1)$. $M$ accepts the input tape $x$ if and only if it eventually halts in an accepting state with all its heads on the bottom boundary symbols #$'$s.

For a device $M$, we denote by $T(M)$ the set of all inputs accepted by $M$. The states of this device are considered to be divided into three disjoint sets of working, accepting, and rejecting states.

A self-verifying nondeterministic device is a device with four types of states: working, accepting, rejecting, and neutral ones. The self-verifying nondeterministic device $M$ is not always to make mistakes. If there is a computation of $M$ on an input $x$ finishing in an accepting (resp., rejecting) state, then $x$ must be in $T(M)$ (resp., $x$ must not be in $T(M)$). For every input $y$, there is at least one computation of $M$ that finishes either in an accepting state (if $y \in T(M)$) or in a rejecting state (if $y \notin T(M)$).

For each $k \geq 1$, let $SV4-kHFA$ denote a seven-way four-dimensional $k$-head finite automaton. In order to represent different kinds of $SV4-kHFA$'s, we use the notation $SV4-XYkHFA$, where

$$X = N : \text{nondeterministic},$$

$$X = SVN : \text{self–verifying nondeterministic};$$

$$Y = SP : \text{simple},$$

there is no $Y : \text{non–simple}$.

We denote by $\mathcal{L}[SV4-XYkHFA]$ the class of sets of input tapes accepted by $SV4-XYkHFA$'s.

Let $r$ be a positive integer. A device $M$ described above is $r$ path-bounded if for any input $x$, there are at most $r$ computation paths of $M$ on $x$. We denote an $r$ path-bounded $SV4-XYkHFA$ by $SV4-XYkHFA(r)$, and denote the class of sets of input tapes accepted by $SV4-XYkHFA(r)$'s by $\mathcal{L}[SV4-XYkHFA(r)]$.

### 3 Non-Simple Case

We first prove a strong separation between $r$ path-bounded and $(r+1)$ path-bounded for seven-way four-dimensional multidhead finite automata.

**Theorem 3.1.** For each positive integers $k \geq 2$ and $r \geq 1$,

$$\mathcal{L}[SV4-SVNkHFA(r+1)] - \mathcal{L}[SV4-NkHFA(r)] \neq \phi.$$

**Proof:** For each positive integers $k \geq 2$ and $r \geq 1$, let $T_1(k,r) = \{ x \in \{ 0, 1 \}^4 | 3 \leq n \geq 2r(k+1) + 1 \text{ } l_1(x) = l_2(x) = l_3(x) = l_4(x) = n \} \land \exists z \in \{ 0, 1 \}^* \exists x \in \{ 0, 1 \}^{l_1(x)+1} \exists y \in \{ 0, 1 \}^{l_2(y)+1} \exists z \in \{ 0, 1 \}^{l_3(z)+1} \exists w \in \{ 0, 1 \}^{l_4(w)+1}$

The symbols forms a line from the first column to the last column in the $(2r(k+1)+1)$th three-dimensional rectangular array of $x$ and from the
first row to the last row in a column one after another), where $b(k) =_{k} C_2$. To prove the theorem, it suffices to show that for each $k \geq 2$ and $r \geq 1$, (1) $T_1(k, r+1) \subseteq L[[SV4-SVnkHFA(r+1)]]$, and (2) $T_1(k, r+1) \not\subseteq L[[SV4-NkHFA(r)]]$. First of all we prove Past (1) of the theorem. $T_1(k, r+1)$ is accepted by an $SV4-SVnkHFA(r+1)$ $M$ which acts as follows. Suppose that an input tape $x$ with $l_1(x) = l_2(x) = l_3(x) = l_4(x) = n$ ($n \geq 2$) is presented to $M$. First, $M$ nondeterministically guesses some $i$ ($0 \leq i \leq r$) and checks whether $x[i] \neq \ast$ and $x[i+1] = \ast$, and from the first row to the last row in a column one after another) for some $z \in \{0, 1\}$, $M$ enters a rejecting state. If $x[i] \neq \ast$ and $x[i+1] = \ast$ (the string of the symbols forms a line from the first column to the last column in the $2(r+1)b(k)+1$th three-dimensional rectangular array of $x$ and from the first row to the last row in a column one after another) for some $z \in \{0, 1\}$, $M$ enters a neutral state, whether or not $x[i+1] = \ast$. This check can easily be done by using a well-known technique in [10]. If $x[i] \neq \ast$ and $x[i+1] = \ast$ (the string of the symbols forms a line from the first row to the last row in the $2(r+1)b(k)+1$th three-dimensional rectangular array of $x$ and from the first row to the last row in a column one after another) for some $z \in \{0, 1\}$, $M$ enters a neutral state, whether or not $x[i+1] = \ast$. This check can easily be done by using a well-known technique in [10].

From Theorem 3.1, we have the following corollary:

**Corollary 3.1.** For each $X \in \{N, SVN\}$, and for each positive integers $k \geq 2$ and $r \geq 1$, $L[[SV4-XnkHFA(r)]] \subseteq L[[SV4-XnHFA(r+1)]]$.

We next show a strong separation between self-verifying nondeterminism and nondeterminism.

**Theorem 3.2.** For each positive integer $k \geq 2$, $L[[SV4-NkHFA(2)]] \not\subseteq L[[SV4-SVnkHFA]]$.

**Proof:** For each positive integer $k \geq 2$, let $T_2(k) = \{ x \in \{0, 1\}^{*} | \exists n \geq 4b(k), \{l_1(x) = l_2(x) = l_3(x) = l_4(x) = n \} \wedge \exists l \geq 0, l \leq n \exists j \leq (i+1)(b(k)+j-1) \}$, where $b(k) =_{k} C_2$. Then, we have $T_2(k) \subseteq L[[SV4-NkHFA(2)]] \subseteq L[[SV4-SVnkHFA]]$. Then, by using the same idea as in [6, 7], we get the desired result.

From Theorems 3.1 and 3.2, we have the following corollary:

**Corollary 3.2.** For each positive integers $k \geq 2$ and $r \geq 1$,

1. $L[[SV4-SVnkHFA]] \not\subseteq L[[SV4-NkHFA]]$,
2. $L[[SV4-SVnkHFA(r)]] \not\subseteq L[[SV4-NkHFA(r)]]$, and
3. $L[[SV4-SVnkHFA(r+1)]]$ and $L[[SV4-NkHFA(r)]]$ are incomparable.

### 4 Simple Case

This section first prove a strong separation between $r$-path-bounded and $(r+1)$-path-bounded machines for the seven-way simple case.

**Theorem 4.1.** For each positive integers $k \geq 2$ and $r \geq 1$, $L[[SV4-SVSPkHFA(r+1)]] \not\subseteq L[[SV4-NSPkHFA(r)]] = \emptyset$.

**Proof:** For each positive integers $k \geq 2$ and $r \geq 1$, let $T_3(k, r) = \{ x \in \{0, 1\}^{*} | \exists n \geq 2r+1, k \} \{ l_1(x) = l_2(x) = l_3(x) = n \} \wedge \{ \{ (1) \}$, $L[[SV4-XSPkHFA]] \not\subseteq L[[SV4-NSPkHFA]]$. Then, by using the same technique as in the proof of Theorem 4.1 in [7], we can get the desired result.

From Theorem 4.1, we have the following corollary:

**Corollary 4.1.** For each $X \in \{N, SVN\}$, and for each positive integers $k \geq 2$ and $r \geq 1$, $L[[SV4-XSPkHFA(r+1)]] \not\subseteq L[[SV4-NSPkHFA(r+1)]]$. We next show a strong separation between self-verifying nondeterminism and nondeterminism.

**Theorem 4.2.** For each positive integer $k \geq 2$, $L[[SV4-XSpkHFA]] \not\subseteq L[[SV4-NSp(1)]]$.

From Theorem 4.2, we have the following corollary:

**Corollary 4.2.** For each positive integer $k \geq 2$, $L[[SV4-XSpkHFA]] \not\subseteq L[[SV4-NSpkHFA]]$. We next show a strong separation between self-verifying nondeterminism and nondeterminism.
\[ \mathcal{L}(SV4-NSPkHFA(2)) = \mathcal{L}(SV4-SVNSPkHFA) \neq \emptyset. \]

**Proof:** For each positive integer \( k \geq 2 \), let \( T_4(k) = \{ x \in \{0, 1\}^4 \mid \exists n \geq \max\{4, k\} \{ l_1(x) = l_2(x) = l_3(x) = l_4(x) = n \} \wedge \exists i (1 \leq i \leq 2) [\text{the } i\text{th three-dimensional rectangular array of } x \text{ has exactly } k \text{ '1's} \wedge x[*, *, *, i] \neq x[*, *, *, i+2]] \} \}. \] Then, by using the standard technique in [6], we can show that

\[ T_4(2k-1) \in \mathcal{L}(SV4-NSPkHFA(2)) - \mathcal{L}(SV4-SVNSPkHFA). \]

From Theorems 4.1 and 4.2, we have the following corollary:

**Corollary 4.2.** For each positive integers \( k \geq 2 \) and \( r \geq 2 \),

1. \( \mathcal{L}(SV4-SVNSPkHFA) \subseteq \mathcal{L}(SV4-NSPkHFA) \),
2. \( \mathcal{L}(SV4-SVNSPkHFA(r)) \subseteq \mathcal{L}(SV4-NSPkHFA(r)) \),

and

3. \( \mathcal{L}(SV4-SVNSPkHFA(r+1)) \) and \( \mathcal{L}(SV4-NSPkHFA(r)) \) are incomparable.

## 5 Conclusion

In this paper, we investigated path-bounded seven-way four-dimensional finite automata, and showed some properties about them. It is interesting to investigate a hierarchy based on the degree of nondeterminism for eight-way four-dimensional multilevel finite automata which can move east, west, south, north, up, down, in the past, or in the future.

**References:**


