PAPER Special Section on Foundations of Computer Science - New Trends of Theory of Computation and Algorithm -

Finite Automata with Colored Accepting States and Their Unmixedness Problems

Yoshiaki TAKAHASHI†a) *and* Akira ITO††b), *Members*

SUMMARY Some textbooks of formal languages and automata theory implicitly state the structural equality of the binary n-dimensional de Bruijn graph and the state diagram of minimum state deterministic finite automaton which accepts regular language $(0+1)^*1(0+1)^{n-1}$. By introducing special finite automata whose accepting states are refined with two or more colors, we extend this fact to both k-ary versions. That is, we prove that k-ary n-dimensional de Brujin graph and the state diagram for minimum state deterministic colored finite automaton which accepts the $(k-1)$-tuple of the regular languages $(0+1+\cdots+k-1)^*1(0+1+\cdots+k-1)^{n-1},\ldots,$ and $(0+1+\cdots+k-1)^*(k-1)(0+1+\cdots+k-1)^{n-1}$ are isomorphic for arbitrary k more than or equal to 2. We also investigate the properties of colored finite automata themselves and give computational complexity results on three decision problems concerning color unmixedness of nondeterminisitic ones.

key words: *de Bruijn graphs, finite automata, state-minimization, NLOG-completeness, NP-completeness, independent set*

1. Introduction

de Bruijn graphs (and their associated sequences) have been used widely in areas of application, such as coding theory, computer network design, and genome assembly in recent years [1]–[12].

One purpose of this paper is to characterize de Bruijn graphs by some regular languages. Here, the characterization of digraphs by languages means that a specific family of graphs is coincident to the graph structure of transition diagrams of finite automata accepting a specific family of regular languages. This claim is validated by the fact [13]–[15] that all state-minimized deterministic finite automata accepting a certain regular language are isomorphic. As an example, Moriya [15] implicitly states that n-dimensional directed hypercube H_n is characterized by the language $L_{\mathrm{e}}^{(n)} = \{x \in \{0,1,\ldots,n-1\}^* \mid$ the number of symbols i's in x is even for each $i = 0,1,\cdots,n-1\}$. That is, H_n and the transition diagram of the minimum state deterministic finite automaton accepting $L_{\mathrm{e}}^{(n)}$ are isomorphic up to edge labeling.

Another implicit example is the isomorphism between binary n-dimensional de Bruijn graph of $DB_{2,n}$ and the transition diagram of minimum state deterministic finite automaton D_n accepting $L_n = \{x \in \{0,1\}^* \mid$ the nth symbol from the right end of x is $1\}$ [16]. The nondeterministic finite automaton N_n used to produce D_n can be easily extended to higher radix k from binary one. However, the correspondent deterministic automaton turns out to be not minimal and shrinks to the binary automaton D_n once we use the well-known minimization algorithm. To get around the situation, we make automata have classifying function of features of input strings into two or more languages in addition to the conventional function of either accepting or non-accepting. We call such a automaton a colored automaton. Colored finite automaton is just a special kind of Moore machines, i.e., finite automata with outputs [5], [17], [18]. While they are input-output transducers, our colored finite automaton remains to be a classifier of input strings into two or more languages and is the least extension of conventional acceptor model. Nondeterministic Moore automaton introduced in [17] by Castiglione et al. is essentially the same acceptor model as ours, although their formalism is based on the sequential machine theory. The relations of ours to their works and others are detailed in the end of Sect. 3.

Based on these preliminaries in Sect. 3, we show in Sect. 4 that k-ary de Bruijn graphs $DB_{k,n}$ and the transition diagram of minimum state deterministic colored finite automaton $D_{k,n}$ accepting the $(k-1)$-tuple of regular languages $(L_{k,n}^{(1)},\ldots,L_{k,n}^{(k-1)})$ are isomorphic, where $L_{k,n}^{(i)} = \{x \in \{0,1,\ldots,k-1\}^* \mid$ the ith symbol from the right end of x is $i\}$.

Secondly in Sect. 5, we investigate the complexity of unmixedness property possibly involved in nondeterministic colored finite automaton (NCFA). This condition might be matter for practitioners who are willing to use colored automata because they should want to obtain unmixed ones unless it is intended. Of course, once we transform the NCFA to deterministic one, the unmixedness becomes apparent. It is well-known that the task of nondeterministic to deterministic transformation of finite automaton, however, consumes exponential time in the worst case. We show that the unmixedness of a given NCFA can be checked in polynomial time. More precisely, it is shown that this problem is *NLOG*-complete. Other problems considered concern to changing an ordinary NFA to an unmixed NCFA. In the case of division problem of accepting states of NFA to k unmixed colors, it is also solved in polynomial time. In the case of extension problem of accepting states of NFA from a single

Manuscript received March 26, 2021.
Manuscript revised July 22, 2021.
Manuscript publicized November 21, 2021.

†The author is with the National Institute of Technology, Oshima College, Yamaguchi-ken, 742–2193 Japan.
††The author is with the Graduate School of Science and Technology for Innovation, Yamaguchi University, Ube-shi, 755–8611 Japan.
a) E-mail: takahashiy@oshima-k.ac.jp
b) E-mail: akito@yamaguchi-u.ac.jp (Corresponding author)

DOI: 10.1587/transinf.2021FCP0012

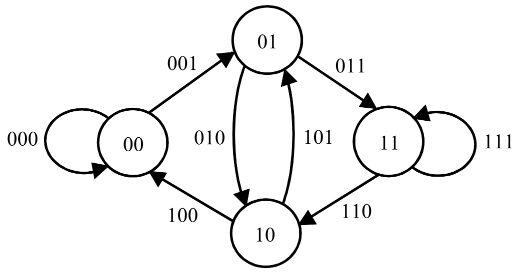

Fig. 1 de Bruijn graph $DB_{2,2}$.

color to k unmixed colors, it is shown to be NP-complete via the reduction from independent set problem of undirected graphs [13], [14], [19].

Readers might doubt that the product of many automata can carry out the same work as colored automata. This is essentially true but an example is shown in Sect. 3 demonstrating that the difference of descriptional complexities is huge. A possible usage of colored automaton is to fully utilize the natural redundancy of a given nondeterministic automaton while keeping the functionality of its product automaton.

2. Definitions and Notations

In this section, we give preliminary definitions and notations [3], [6], [8], [9], [13], [14], [14], [20].

Definition 1: A 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$ defined as follows is called nondeterministic finite automaton and abbreviated NFA.

1. Q is a finite set of states,
2. Σ is a finite set of input symbols,
3. δ is the transition function from $Q \times \Sigma$ to 2^Q,
4. $q_0 \in Q$ is the initial state,
5. $F \subseteq Q$ is the set of accepting states.

If each $\delta(q, a)$ is a set with exactly one element, M is called deterministic and abbreviated DFA. When M starts from the initial state q_0 and finishes after it reads the input string x, we say that x is accepted by M if its final state is in F. We define the language accepted by M as $L(M) \stackrel{\triangle}{=} \{x \in \Sigma^* \mid x \text{ is accepted by } M\}$[†].

Fact 1 (Subset construction method): For an NFA $M = (Q, \Sigma, \delta, q_0, F)$, let DFA $M' = (2^Q, \Sigma, \delta', \{q_0\}, \{S \subseteq Q \mid S \cap F \neq \emptyset\})$, where $\delta'(S, a) = \bigcup_{p \in S} \delta(p, a)$ for each $S \subseteq Q, a \in \Sigma$, then $L(M') = L(M)$.

Definition 2: Directed graph defined as follows is called k-ary n-dimensional de Bruijn graph and abbreviated $DB_{k,n}$.

$$\begin{cases} V = \{b_1 b_2 \cdots b_n \mid b_i \in \{0, 1, \ldots, k-1\}, i = 1, \ldots, n\}, \\ E = \{(b_1 b_2 \cdots b_n, b'_1 b'_2 \cdots b'_n) \mid \\ \quad b_i, b'_i \in \{0, 1, \ldots, k-1\}, i = 1, \ldots, n, \\ \quad b_2 = b'_1, b_3 = b'_2, \ldots, b_n = b'_{n-1}\}. \end{cases}$$

[†]X $\stackrel{\triangle}{=}$ Y means that X is defined as Y.

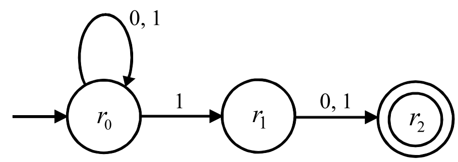

Fig. 2 NFA N_2 accepting L_2.

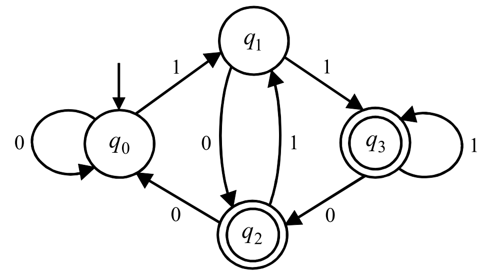

Fig. 3 DFA D_2 accepting L_2.

$DB_{2,2}$ is shown in Fig. 1. In the figure, edge labels of binary digits indicate their starting nodes and end nodes.

We consider the language $L_n = \{x \in \{0, 1\}^* \mid$ the nth symbol from the end of x is 1 $\}$, i.e., the set of strings over $\{0,1\}$ whose nth symbols from their right ends are 1's. An NFA N_n accepting L_n is as follows.
For each $i = 1, \ldots, n-1$ and $a \in \{0, 1\}$,

$$N_n = (\{r_0, r_1, \ldots, r_n\}, \{0, 1\}, \delta, r_0, \{r_n\}),$$
$$\delta(r_0, 0) = \{r_0\}, \delta(r_0, 1) = \{r_0, r_1\}, \delta(r_i, a) = \{r_{i+1}\},$$
$$i = 1, \ldots, n-1, a \in \{0, 1\}.$$

Figure 2 is the transition diagrams of N_2. Furthermore, D_2 obtained from N_2 using subset construction method is shown in Fig. 3.

3. Colored Finite Automata

In this section, we introduce colored finite automata and investigate their fundamental properties.

Definition 3: Let L_i be a language over some alphabet Σ for $i = 1, \ldots, k, k \geq 1$. (1) k-tuple $(L_1, L_2, \cdots, L_k)$ of languages is called *colored language* (vector) of k colors over Σ. (2) If a language L is expressed with the direct sum[††] $\sum_{i=1}^{k} L_i$ of these languages, L is called *distinctly colored language* of k colors over Σ.

The above terminology language vector or tuple of languages may sound strange but the same concept is implicitly used in some field of formal grammars [15], [21], [22]. For example, during the derivation process of a terminal string for a multiple context-free grammar [21], which is a slight extension of context-free grammar, a multi-dimensional

[††]For sets X and Y, direct sum X + Y is the union X ∪ Y satisfying the disjointness X ∩ Y = ∅. Notice that we use the regular expression $r + s$ to denote language $L_1 \cup L_2$, where L_1 and L_2 are the languages expressed by the regular expressions r and s, respectively.

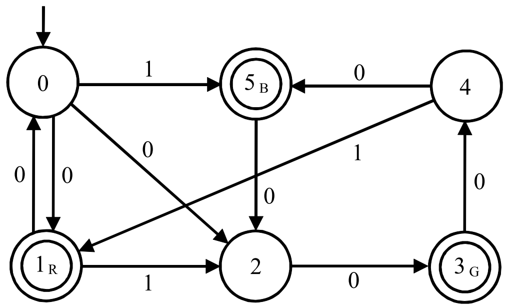

Fig. 4 An example M_0 of NCFA.

vector of sentential forms is rewritten to another one of a different dimensionality.

In our case of colored automaton, its input strings are vectorized or classified with colors just after it enters the accepting states with the corresponding colors.

Definition 4: A 5-tuple $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^{k} F_i)$ as follows is called *nondeterministic colored finite automaton* and abbreviated NCFA.

1. Q is a finite set of states,
2. Σ is a finite set of input symbols,
3. δ is the transition function from $Q \times \Sigma$ to 2^Q,
4. $q_0 \in Q$ is the initial state,
5. $\sum_{i=1}^{k} F_i \subseteq Q$ is the set of colored accepting states, where F_i is the set of accepting states with ith color.

The following example is intended to make it easy to understand the concepts introduced here and demonstrate usefulness of our results. Readers confused by the nontriviality are recommended to apply the concepts of this paper to other familiar NFA examples found in standard textbooks [13], [14], [23].

Example 1: An example $M_0 = (Q, \Sigma, \delta, q_0, \Sigma_{i=1}^{3} F_i)$ of NCFA is shown in Fig. 4, where

$$Q = \{0, 1_{\mathrm{R}}, 2, 3_{\mathrm{G}}, 4, 5_{\mathrm{B}}\},$$
$$\Sigma = \{0, 1\},$$
$$q_0 = 0,$$
$$\Sigma F_i = F_{\mathrm{R}} + F_{\mathrm{G}} + F_{\mathrm{B}}, F_{\mathrm{R}} = \{1_{\mathrm{R}}\}, F_{\mathrm{G}} = \{3_{\mathrm{G}}\}, F_{\mathrm{B}} = \{5_{\mathrm{B}}\}.$$

■

If each $\delta(q, a)$ is a set with exactly one element, M is called deterministic and abbreviated DCFA.

We denote as $\hat{\delta}(q, x)$ the set of reachable states when M starts from state q and finishes after it reads the input string x. If $\hat{\delta}(q, x) \cap F_i \neq \emptyset$, we say that M accepts x with ith color.

$$L_i(M) \triangleq \{x \in \Sigma^* \mid \hat{\delta}(q_0, x) \cap F_i \neq \emptyset\}$$

is called the language accepted by M with ith color and

$$L(M) \triangleq \bigcup_{i=1}^{k} L_i(M)$$

is called the (unified) language accepted by M. Especially, if it holds that

$$L(M) = \sum_{i=1}^{k} L_i(M),$$

we say that $L(M)$ is *unmixed* and that M *color-distinctly* accepts $L(M)$. Note that when M is deterministic or $k = 1$, it is inherently unmixed.

Example 2: Consider the same NCFA as in Example 1. Then, for examples,

$$L_{\mathrm{R}}(M_0) = (00 + (01 + 0 + 10)(0000 + 0011)^*0010)^*$$
$$\cdot ((01 + 0 + 10)(000 + 0011)^*001 + 0),$$

$$\vdots$$

$$L(M_0) = L_{\mathrm{R}}(M_0) \cup L_{\mathrm{G}}(M_0) \cup L_{\mathrm{B}}(M_0)$$
$$= (00 + (01 + 0 + 10(0000 + 0011)^*0010)^*$$
$$\cdot ((01 + 0 + 10)(0000 + 0011)^*$$
$$\cdot (00 + 0 + 001) + 0 + 1).$$

These are obtained by using well-known translation method from NFAs to regular expressions, taking M_0 as separated ordinary NFAs $M_0^{\mathrm{R}}, M_0^{\mathrm{G}}, M_0^{\mathrm{B}}$, and M_0^{RGB} whose accepting states are $F_{\mathrm{R}}, F_{\mathrm{G}}, F_{\mathrm{B}}$, and $F_{\mathrm{R}} \cup F_{\mathrm{G}} \cup F_{\mathrm{B}}$, respectively. ■

For any $x \in \Sigma^*$, there exists a unique $I \subseteq \{1, \ldots, k\}$ such that $x \in L_i(M), i \in I, x \notin L_j(M), j \notin I$. In other words,

$$I(x) \triangleq \{i \in \{1, \ldots, k\} \mid \hat{\delta}(q_0, x) \cap F_i \neq \emptyset\}$$

is a mapping $I : \Sigma^* \to 2^{\{1,\ldots,k\}}$.

The following fact is obvious from the definitions.

Fact 2: (1) $x \in L_i(M) \Leftrightarrow i \in I(x)$.
(2) $x \in L(M) \Leftrightarrow \exists i \in \{1, \ldots, k\}[i \in I(x)] \Leftrightarrow I(x) \neq \emptyset$.

Proposition 1: $L(M)$ is unmixed $\Leftrightarrow$ For any $x \in \Sigma^*$, $|I(x)| \leq 1$.

(Proof)

$L(M)$ is mixed
$$\Leftrightarrow \exists i, j, i \neq j \, \exists x \in \Sigma^*[x \in L_i(M) \cap L_j(M)]$$
$$\Leftrightarrow \exists i, j \, \exists x \in \Sigma^*[i, j \in I(x), i \neq j]$$
$$\Leftrightarrow \exists x \in \Sigma^*[|I(x)| \geq 2].$$

□

For each $I \subseteq \{1, \ldots, k\}$, define

$$F'_I \triangleq \{S \subseteq Q \mid S \cap F_i \neq \emptyset, i \in I, S \cap F_j = \emptyset, j \notin I\}.$$

That is, F'_I is the set of state subsets each of which contains accepting states with ith color belonging to I but does not contain accepting states with jth color not belonging to I.

Proposition 2: (1) If $I \neq J$, $F'_I \cap F'_J = \emptyset$.

(2) $$\bigcup_{I \subseteq \{1,\ldots,k\}, I \neq \emptyset} F'_I = \bigcup_{i=1}^{k} \{S \subseteq Q \mid S \cap F_i \neq \emptyset\}.$$

(Proof)

(1) Suppose $S \in F'_I \cap F'_J$ to the contrary. Without loss of generality, let $i_0 \in I, i_0 \notin J$, then

$$\begin{aligned} & S \in F'_I, S \in F'_J \\ & \quad \Leftrightarrow S \cap F_i \neq \emptyset, i \in I, S \cap F_j = \emptyset, j \notin I, \\ & \qquad S \cap F_j \neq \emptyset, j \in J, S \cap F_i = \emptyset, i \notin J \\ & \quad \Rightarrow S \cap F_{i_0} \neq \emptyset, S \cap F_{i_0} = \emptyset. \end{aligned}$$

This is a contradiction.

(2) For any $S \subseteq Q$, there exists a unique $I \subseteq \{1, \ldots, k\}$ such that $S \cap F_i \neq \emptyset, i \in I, S \cap F_j = \emptyset, j \notin I$. In other words,

$$I(S) \triangleq \{i \in \{1, \ldots, k\} \mid S \cap F_i \neq \emptyset\}$$

is a mapping $I : 2^Q \to 2^{\{1,\cdots,k\}}$. From

$$F'_I = \{S \subseteq Q \mid I(S) = I\},$$

we have

$$\begin{aligned} & \cup_{i=1}^k \{S \subseteq Q \mid S \cap F_i \neq \emptyset\} \\ & \quad = \{S \subseteq Q \mid \exists i \in \{1, \ldots, k\}[S \cap F_i \neq \emptyset]\} \\ & \quad = \{S \subseteq Q \mid \exists I \subseteq \{1, \ldots, k\} I \neq \emptyset, [I(S) = I]\} \\ & \quad = \cup_{I \subseteq \{1,\ldots,k\}, I \neq \emptyset} F'_I. \end{aligned}$$

□

Example 3: Consider the same NCFA M_0 as in Example 1. It is easily verified that

$$\mathrm{G}, \mathrm{B} \in I(0000), \text{ so } |I(0000)| \geq 2.$$

Thus, $L(M_0)$ is mixed. From

$$\begin{aligned} F'_\emptyset &= \{S \subseteq Q \mid S \cap F_\mathrm{R} = \emptyset, S \cap F_\mathrm{G} = \emptyset, S \cap F_\mathrm{B} = \emptyset\} \\ &= \{\{0\}, \{2\}, \{4\}, \emptyset, \{0, 2\}, \{0, 4\}, \{2, 4\}, \{0, 2, 4\}\}, \\ F'_{\{\mathrm{R}\}} &= \{S \subseteq Q \mid S \cap F_\mathrm{R} \neq \emptyset, S \cap F_\mathrm{G} = \emptyset, S \cap F_\mathrm{B} = \emptyset\} \\ &= \{\{1_\mathrm{R}\}, \{1_\mathrm{R}, 0\}, \{1_\mathrm{R}, 2\}, \{1_\mathrm{R}, 4\}, \{1_\mathrm{R}, 0, 2\}, \\ &\quad \{1_\mathrm{R}, 0, 4\}, \{1_\mathrm{R}, 2, 4\}, \{1_\mathrm{R}, 0, 2, 4\}\}, \end{aligned}$$

$$\vdots$$

$$\begin{aligned} F'_{\{\mathrm{RG}\}} &= \{S \subseteq Q \mid S \cap F_\mathrm{R} \neq \emptyset, S \cap F_\mathrm{G} \neq \emptyset, S \cap F_\mathrm{B} = \emptyset\} \\ &= \{\{1_\mathrm{R}, 3_\mathrm{G}\}, \{1_\mathrm{R}, 3_\mathrm{G}, 0\}, \{1_\mathrm{R}, 3_\mathrm{G}, 2\}, \{1_\mathrm{R}, 3_\mathrm{G}, 4\}, \\ &\quad \{1_\mathrm{R}, 3_\mathrm{G}, 0, 2\}, \{1_\mathrm{R}, 3_\mathrm{G}, 0, 4\}, \{1_\mathrm{R}, 3_\mathrm{G}, 2, 4\}, \\ &\quad \{1_\mathrm{R}, 3_\mathrm{G}, 0, 2, 4\}\}, \end{aligned}$$

$$\vdots$$

$$\begin{aligned} F'_{\{\mathrm{RGB}\}} &= \{S \subseteq Q \mid S \cap F_\mathrm{R} \neq \emptyset, S \cap F_\mathrm{G} \neq \emptyset, S \cap F_\mathrm{B} \neq \emptyset\} \\ &= \{\{1,_\mathrm{R}, 3_\mathrm{G}, 5_\mathrm{B}\}, \{1_\mathrm{R}, 3_\mathrm{G}, 5_\mathrm{B}, 0\}, \{1_\mathrm{R}, 3_\mathrm{G}, 5_\mathrm{B}, 2\}, \\ &\quad \{1_\mathrm{R}, 3_\mathrm{G}, 5_\mathrm{B}, 4\}, \{1_\mathrm{R}, 3_\mathrm{G}, 5_\mathrm{B}, 0, 2\}, \\ &\quad \{1_\mathrm{R}, 3_\mathrm{G}, 5_\mathrm{B}, 0, 4\}, \{1_\mathrm{R}, 3_\mathrm{G}, 5_\mathrm{B}, 2, 4\}, \\ &\quad \{1_\mathrm{R}, 3_\mathrm{G}, 5_\mathrm{B}, 0, 2, 4\}\}, \end{aligned}$$

we have

$$\begin{aligned} \bigcup_{I \subseteq \{R,G,B\}, I \neq \emptyset} F'_I &= \sum_{I \subseteq \{R,G,B\}, I \neq \emptyset} F'_I \\ &= F'_{\{\mathrm{R}\}} + F'_{\{\mathrm{G}\}} + F'_{\{\mathrm{B}\}} + F'_{\{\mathrm{RG}\}} + F'_{\{\mathrm{RB}\}} \\ &\quad + F'_{\{\mathrm{GB}\}} + F'_{\{\mathrm{RGB}\}}. \end{aligned}$$

■

Theorem 1 (Subset construction method for NCFA):
For an NCFA

$$M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^k F_i),$$

let DCFA

$$M' = (2^Q, \Sigma, \delta', \{q_0\}, \sum_{I \subseteq \{1,\ldots,k\}, I \neq \emptyset} F'_I),$$

where

$$\delta'(S, a) = \bigcup_{p \in S} \delta(p, a), S \subseteq Q, a \in \Sigma.$$

Then, by defining $L_I(M') \triangleq \{x \in \Sigma^* \mid \hat{\delta}'(\{q_0\}, x) \in F'_I\}$ and $F'[\{q_0\}] \triangleq \{S \subseteq Q \mid \exists x \in \Sigma^*[S = \hat{\delta}'(\{q_0\}, x), S \in \sum_{I \subseteq \{1,\ldots,k\}, I \neq \emptyset} F'_I]\}$, we have the following.

(1) $L_I(M') = \bigcap_{i \in I} L_i(M) - \bigcup_{j \notin I} L_j(M), I \subseteq \{1, \ldots, k\}$,
i.e., each individual language of M' truly reflects the mixedness situation of M.

(2) $L(M') = \sum_{I \subseteq \{1,\ldots,k\}, I \neq \emptyset} L_I(M') = L(M)$,
i.e., the unified languages accepted by both M and M' coincide as a whole.

(3) $L(M)$ is unmixed

$$\Leftrightarrow L_i(M) = L_{\{i\}}(M') \text{ for each } i \in \{1, \ldots, k\},$$

$$\Leftrightarrow F'[\{q_0\}] \subseteq \sum_{i=1}^k F'_{\{i\}},$$

i.e., in the unmixed case the number of colors and colored languages of M' are the same as M.

(Proof)

(1) $x \in \cap_{i \in I} L_i(M) - \cup_{j \notin I} L_j(M)$

$$\begin{aligned} & \Leftrightarrow x \in L_i(M), i \in I, x \notin L_j(M), j \notin I \\ & \Leftrightarrow \hat{\delta}(q_0, x) \cap F_i \neq \emptyset, i \in I, \hat{\delta}(q_0, x) \cap F_j = \emptyset, j \notin I. \end{aligned}$$

Now, it holds that $\hat{\delta}(q_0, x) = \hat{\delta}'(\{q_0\}, x)$ because the NCFA version of subset construction method is the same as ordinary NFA version except its accepting states. Note that the left part of the equation represents a set of NFA's states and the right part represents one of DFA's states. Thus,

$$\begin{aligned} & \text{the above predicate} \\ & \quad \Leftrightarrow \hat{\delta}'(\{q_0\}, x) \cap F_i \neq \emptyset, i \in I, \\ & \qquad \hat{\delta}'(\{q_0\}, x) \cap F_j = \emptyset, j \notin I \end{aligned}$$

$\Leftrightarrow \hat{\delta}'(\{q_0\}, x) \in F'_I$
$\Leftrightarrow x \in L_I(M')$.

(2) Supposing $x \in L_I(M') \cap L_J(M'), I \neq J$ to the contrary, we have $\hat{\delta}'(\{q_0\}, x) \in F'_I \cap F'_J, I \neq J$, which contradicts $\bigcup F_I = \sum F_I$. Therefore, $\bigcup L_I(M') = \sum L_I(M')$.

$$
\begin{aligned}
& x \in L(M') \\
& \quad \Leftrightarrow \exists I \subseteq \{1, \ldots, k\}, I \neq \emptyset[x \in L_I(M')] \\
& \quad \Leftrightarrow x \notin L_\emptyset(M') \\
& \quad \Leftrightarrow \hat{\delta}'(\{q_0\}, x) \notin F'_\emptyset \\
& \quad \Leftrightarrow \overline{\forall j \in \{1, \ldots, k\}[\hat{\delta}'(\{q_0\}, x) \cap F_j = \emptyset]} \\
& \quad \Leftrightarrow \exists j \in \{1, \ldots, k\}[\hat{\delta}'(\{q_0\}, x) \cap F_j \neq \emptyset] \\
& \quad \Leftrightarrow \exists j \in \{1, \ldots, k\}[x \in L_j(M)] \\
& \quad \Leftrightarrow x \in L(M),
\end{aligned}
$$

where $\overline{\mathrm{P}}$ denotes the negation of predicate P.

(3) From the part (1) of this Theorem,

$$
\begin{aligned}
& L(M) \text{ is unmixed} \\
& \quad \Leftrightarrow L_i(M) \cap L_j(M) = \emptyset \text{ for any } i, j, i \neq j \\
& \quad \Leftrightarrow L_{\{i\}}(M') = L_i(M) - \cup_{j \neq i} L_j(M) \\
& \qquad = L_i(M) - \cup_{j \neq i}(L_i(M) \cap L_j(M)) \\
& \qquad = L_i(M) \text{ for any } i.
\end{aligned}
$$

For the last equivalence,

$$
\begin{aligned}
& L(M) \text{ is mixed} \\
& \quad \Leftrightarrow \exists x \in \Sigma^*[|I(x)| \geq 2] \\
& \quad \Leftrightarrow \exists x \in \Sigma^* \exists I \subseteq \{1, \ldots, k\}, |I| \geq 2 \\
& \qquad [\hat{\delta}(q_0, x) \cap F_i \neq \emptyset, i \in I, \hat{\delta}(q_0, x) \cap \\
& \qquad F_j = \emptyset, j \notin I] \\
& \quad \Leftrightarrow \exists x \in \Sigma^* \exists I \subseteq \{1, \ldots, k\}, \\
& \qquad |I| \geq 2[\hat{\delta}'(\{q_0\}, x) \in F'_I] \\
& \quad \Leftrightarrow \exists I \subseteq \{1, \ldots, k\}, \\
& \qquad |I| \geq 2 \exists x \in \Sigma^*[\hat{\delta}'(\{q_0\}, x) \in F'_I].
\end{aligned}
$$

Thus,

$L(M)$ is unmixed
$\quad \Leftrightarrow \forall I \subseteq \{1, \ldots, k\}, |I| \geq 2,$
$\qquad \forall x \in \Sigma^*[\hat{\delta}'(\{q_0\}, x) \notin F'_I]$
$\quad \Leftrightarrow$ any accepting state of M' reachable from the initial state $\{q_0\}$ belongs to F'_I such that $|I| \leq 1$.
$\quad \Leftrightarrow F'[\{q_0\}] \subseteq \Sigma_{i=1}^k F'_{\{i\}} \subseteq \Sigma_{I \subseteq \{1,\ldots,k\}, I \neq \emptyset} F'_I$.

□

Note that in addition to the exponential blow-up of the size of states, the number of colors of M' could blow up exponentially too, i.e., from k to $2^k - 1$.

Example 4: Figure 5 shows the DCFA M_1 converted from NCFA M_0 by using subset construction method for NCFA, where

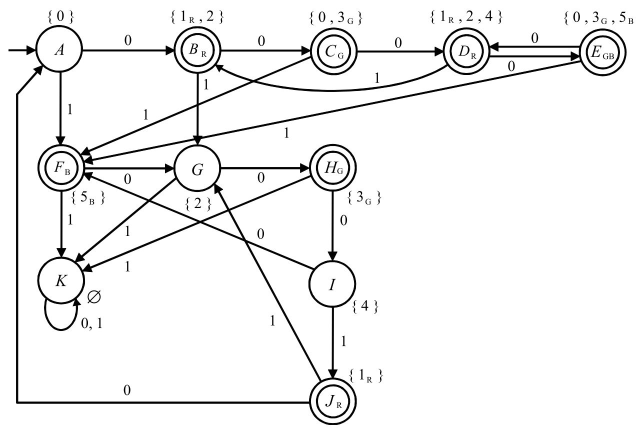

Fig. 5 DCFA M_1 constructed from NCFA M_0.

$$
\begin{aligned}
& M_1 = (Q, \Sigma, \delta, q_0, \Sigma_{I \subseteq \{\mathrm{RGB}\}, I \neq \emptyset} F'_I), \\
& Q = \{A, B_\mathrm{R}, C_\mathrm{G}, D_\mathrm{R}, E_\mathrm{GB}, F_\mathrm{B}, G, H_\mathrm{G}, I, J_\mathrm{R}, K\}, \\
& \Sigma = \{0, 1\}, \\
& q_0 = A, \\
& \Sigma F'_I = F'_{\{\mathrm{R}\}} + F'_{\{\mathrm{G}\}} + F'_{\{\mathrm{B}\}} + F'_{\{\mathrm{GB}\}}, F'_{\{\mathrm{R}\}} = \{B_\mathrm{R}, D_\mathrm{R}, J_\mathrm{R}\}, \\
& F'_{\{\mathrm{G}\}} = \{C_\mathrm{G}, H_\mathrm{G}\}, F'_{\{\mathrm{B}\}} = \{F_\mathrm{B}\}, F'_{\{\mathrm{GB}\}} = \{E_\mathrm{GB}\}.
\end{aligned}
$$

Note that M_1 has four different colors, increased by one combination color GB from the original three colors R, G, and B of M_0. ■

As the final remark of this section, we refer the relationship of colored finite automata to other formalisms. In [17], the authors introduced the same concept of unmixedness, so-called semi-coherency. Their treatment of semi-coherent finite automata is different from ours in the following sense: (1) Output color of each accepting state of deterministic finite automaton converted by subset construction from nondeterministic finite automaton is defined only if it is semi-coherent (otherwise it is undefined). Note that the resulting DCFA converted by our naive subset construction reflects the mixedness of the original NCFA literally. (2) Coherency of nondeterministic finite automaton can be checked only after the conversion to deterministic one, which consumes exponential time in the worst case, sharply contrasted with our polynomial time algorithm shown in Sect. 5.

Unmixedness is a prerequisite of self-verifying finite automata [24], which cannot enter a Yes-colored accepting state and a No-colored (rejecting) state simultaneously. A measure to avoid the mixedness situation is to give a certain order structure to the set of colors, such as a lattice [25], semi-ring [26], [27], etc. and automatically select a unique color of highest priority among accepting colors.

Our approach is to admit the mixedness of colors and the multi-dimensionality of languages but wish to decrease them as much as possible.

4. Equivalence of $DB_{k,n}$ and State-Minimized and Colored Finite Automaton $D_{k,n}$

In this section, we show that the graph structure of a certain deterministic colored finite automaton is isomorphic to k-ary de Bruijn graph of n-dimensional $DB_{k,n}$.

Define

$$N_{k,n} = (Q, \{0, 1, \ldots, k-1\}, \delta, r_0, \sum_{i=1}^{k-1} F_i),$$

where

$$\begin{aligned}
&Q = \{r_0, r_{11}, \ldots, r_{1n}, \cdots, r_{(k-1)1}, \ldots, r_{(k-1)n}\},\\
&\delta(r_0, 0) = \{r_0\},\\
&\delta(r_0, a) = \{r_0, r_{a1}\} \text{ for each } a \in \{0, 1, \ldots, k-1\},\\
&\delta(r_{ij}, a) = \{r_{ij+1}\} \text{ for each } i = 1, \ldots, k-1,\\
&\qquad a \in \{0, 1, \ldots, k-1\},\\
&F_i = \{r_{in}\} \text{ for each } i = 1, \ldots, k-1.
\end{aligned}$$

Figure 6 illustrates the transition diagram of general $N_{k,n}$. It is clear that $N_{k,n}$ is unmixed and

$$\begin{aligned}
L(N_{k,n}) &= \{x \in \{0, 1, \ldots, k-1\}^* \mid \text{the } n\text{th symbol}\\
&\qquad \text{from the end of } x \text{ is either } 1, \ldots, \text{ or } k-1\}\\
&= (0 + 1 + \cdots + k-1)^*(1 + \cdots + k-1)\\
&\qquad (0 + 1 + \cdots + k-1)^{n-1}\\
&= \sum_{i=1}^{k-1} L_i(N_{k,n}),
\end{aligned}$$

where

$$\begin{aligned}
L_i(N_{k,n}) &= \{x \in \{0, 1, \ldots, k-1\} \mid \text{the } n\text{th symbol}\\
&\qquad \text{from the end of } x \text{ is } i\}\\
&= (0 + 1 + \cdots + k-1)^* i\\
&\qquad (0 + 1 + \cdots + k-1)^{n-1},
\end{aligned}$$

for each $i = 1, \ldots, k-1$. In the following, we abbreviate

$$L_{k,n} \triangleq L(N_{k,n}) \text{ and } L^{(i)}_{k,n} \triangleq L_i(N_{k,n})$$

for each $i = 1, \ldots, k-1$.

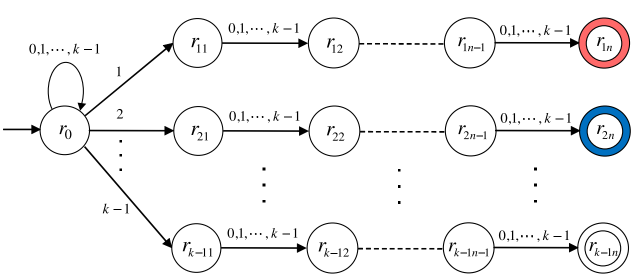

Fig. 6 NCFA $N_{k,n}$ accepting $L_{k,n}$.

Theorem 2: DCFA $D_{k,n}$ constructed from $N_{k,n}$ by using subset construction method for NCFA isomorphic to $DB_{k,n}$ for any $k \geq 2$, $n \geq 1$.

(Proof) In the following, we denote

$$\mathbf{B}^{(k)} = \{\mathbf{0}, \mathbf{1}_1, \mathbf{1}_2, \ldots, \mathbf{1}_{k-1}\},$$

where

$$\mathbf{0} = [\overbrace{00\cdots0}^{k-1}]^T, \mathbf{1}_i = [\overbrace{0\cdots0}^{i-1}1\overbrace{00\cdots0}^{k-i}]^T, i = 1, \ldots, k-1.$$

Applying the NCFA version of subset construction method to $N_{k,n}$, we get the following DCFA $D_{k,n}$.

$$D_{k,n} = (Q', \{0, 1, \ldots, k-1\}, \delta', q'_0, \sum_{i=1}^{k-1} F'_{\{i\}}),$$

where

$$\begin{aligned}
&Q' = \{[1\mathbf{x}_1 \cdots \mathbf{x}_n] \mid \mathbf{x}_j \in \mathbf{B}^{(k)}, j = 1, \ldots, n\},\\
&q'_0 = [1\mathbf{0}\cdots\mathbf{0}],\\
&F'_{\{i\}} = \{[1\mathbf{x}_1 \cdots \mathbf{x}_{n-1}\mathbf{1}_i] \mid \mathbf{x}_j \in \mathbf{B}^{(k)},\\
&j = 1, \ldots, n-1\}.
\end{aligned}$$

For each $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbf{B}^{(k)}, a \in \{0, 1, \ldots, k-1\}$,

$$\delta'([1\mathbf{b}_1 \cdots \mathbf{b}_n], a) = \begin{cases} [1\mathbf{0}\mathbf{b}_1 \cdots \mathbf{b}_{n-1}], & \text{if } a = 0,\\ [1\mathbf{1}_1\mathbf{b}_1 \cdots \mathbf{b}_{n-1}], & \text{if } a = 1,\\ \quad\vdots & \\ [1\mathbf{1}_{k-1}\mathbf{b}_1 \cdots \mathbf{b}_{n-1}], & \text{if } a = k-1, \end{cases}$$

where $[1\mathbf{b}_1 \cdots \mathbf{b}_n]$ denotes 0-1 sequence (characteristic function) which represents a subset of $Q = \{r_0, r_{11}, \ldots, r_{(k-1)1}, \ldots, r_{1n}, \ldots, r_{(k-1)n}\}$.

Note that the state transition of $D_{k,n}$ corresponds to $k-1$ vertically connected n-stage shift registers whose input is $k-1$ bits unary expression of symbol $0, 1, \cdots$, or $k-1$.

The above derivation is clearly seen by the following claim.

Claim 1: At the j-step of subset construction method based on the breadth-first search, which begins to search from the initial state and searches states with smaller distances from the initial state earlier than longer ones, the state vector is expressed with

$$[1\mathbf{b}_1 \cdots \mathbf{b}_j\mathbf{0}\cdots\mathbf{0}], j = 0, 1, \ldots, n.$$

(Proof) The initial state $q'_0 = \{q_0\} = [1\mathbf{0}\cdots\mathbf{0}]$ is obvious. If the state set vector of M' generated in the j-step is $[1\mathbf{b}_1 \cdots \mathbf{b}_j\mathbf{0}\cdots\mathbf{0}]$, then from

$$\delta'([1\mathbf{b}_1 \cdots \mathbf{b}_j\mathbf{0}\cdots\mathbf{0}], a) = \begin{cases} [1\mathbf{0}\mathbf{b}_1 \cdots \mathbf{b}_j\mathbf{0}\cdots\mathbf{0}], & \text{if } a = 0,\\ [1\mathbf{1}_1\mathbf{b}_1 \cdots \mathbf{b}_j\mathbf{0}\cdots\mathbf{0}], & \text{if } a = 1,\\ \quad\vdots & \\ [1\mathbf{1}_{k-1}\mathbf{b}_1 \cdots \mathbf{b}_j\mathbf{0}\cdots\mathbf{0}], & \text{if } a = k-1, \end{cases}$$

for each $a \in \{0, 1, \ldots, k-1\}$, the vector of $(j+1)$-step is expressed with $[1\mathbf{b}'_1\mathbf{b}'_2 \cdots \mathbf{b}'_{j+1}\mathbf{0} \cdots \mathbf{0}]$, where $\mathbf{b}'_1 \in \mathbf{B}^{(k)}, \mathbf{b}'_2 = \mathbf{b}_1, \ldots, \mathbf{b}'_{j+1} = \mathbf{b}_j$. Especially, when $j = n$, it is expressed with $[1\mathbf{b}_1\mathbf{b}_2 \cdots \mathbf{b}_n]$. □

(The proof of Theorem 2 continued) By the left / right inversion of $\mathbf{b}_1 \cdots \mathbf{b}_n$ and omission of the left most 1, we can rewrite the description of $D_{k,n}$ to the following,

$$Q' = \{[\mathbf{x}_n \cdots \mathbf{x}_1] \mid \mathbf{x}_j \in \mathbf{B}^{(k)}, j = 1, \ldots, n\},$$
$$q'_0 = [\mathbf{0} \cdots \mathbf{0}],$$
$$F'_{\{i\}} = \{[\mathbf{1}_i\mathbf{x}_{n-1} \cdots \mathbf{x}_1] \mid \mathbf{x}_j \in \mathbf{B}^{(k)},\ j = 1, \ldots, n-1\},$$
$$\delta'([\mathbf{b}_n \cdots \mathbf{b}_1], a) = \begin{cases} [\mathbf{b}_{n-1} \cdots \mathbf{b}_1\mathbf{0}], & \text{if } a = 0, \\ [\mathbf{b}_{n-1} \cdots \mathbf{b}_1\mathbf{1}_1], & \text{if } a = 1, \\ \quad \vdots & \\ [\mathbf{b}_{n-1} \cdots \mathbf{b}_1\mathbf{1}_{k-1}], & \text{if } a = k-1. \end{cases}$$

Furthermore, regarding vertical vectors $\mathbf{0}, \mathbf{1}_1, \mathbf{1}_2, \ldots,$ and $\mathbf{1}_{k-1}$ as $0, 1, 2, \ldots,$ and $k-1$ in k-ary numeral, respectively, we can rewrite the description of $D_{k,n}$ to the following.

$$Q' = \{[x_n \cdots x_1]_k \mid 0 \le x_j < k, j = 1, \ldots, n\}$$
$$= \{q_0, \ldots, q_{k^n-1}\},$$
$$q'_0 = q_0 = [0 \cdots 0]_k.$$

For each $i = 1, \ldots, k-1$,

$$F'_{\{i\}} = \{[ix_{n-1} \cdots x_1]_k \mid 0 \le x_j < k, j = 1, \cdots, n-1\}$$
$$= \{q_{ik^{n-1}}, \cdots, q_{(i+1)k^{n-1}-1}\}.$$

For each $i = 0, \ldots, k^n - 1, a \in \{0, 1, \ldots, k-1\}$,

$$\delta'(q_i, a) = \begin{cases} q_{ki \bmod k^n}, & \text{if } a = 0, \\ q_{(ki+1) \bmod k^n}, & \text{if } a = 1, \\ \quad \vdots & \\ q_{(ki+k-1) \bmod k^n}, & \text{if } a = k-1. \end{cases}$$

The above description of $D_{k,n}$ is identical to the description of $DB_{k,n}$ in Definition 2:

$$\begin{cases} V = \{0, 1, \ldots, k^n - 1\}, \\ E = \{(x, (kx+i) \bmod k^n) \mid x \in V, i = 0, \ldots, k-1\}. \end{cases}$$

□

Figure 7 shows the transition diagrams of $D_{3,2}$.

Note that the number of states of nondeterministic finite automaton $N_{k,n}$ increases from $|Q| = (k-1)n + 1$ to $|Q'| = k^n$ of deterministic finite automaton $D_{k,n}$ and the number of states of the product automaton of $k-1$ noncolored DFAs $D_{2,n}$'s is $(2^n)^{k-1} = 2^{(k-1)n}$, which is an exponential function concerning k.

The following fact shows that the set of states of DCFA $D_{k,n}$ constructed by subset construction method cannot be reduced any more.

Fact 3: Any DCFA which color-distinctly accepts the distinctly colored language $L_{k,n} = \sum_{i=1}^{k} L_{k,n}^{(i)}$ requires more than or equal to k^n states, where for each $i = 1, \ldots, k-1$,

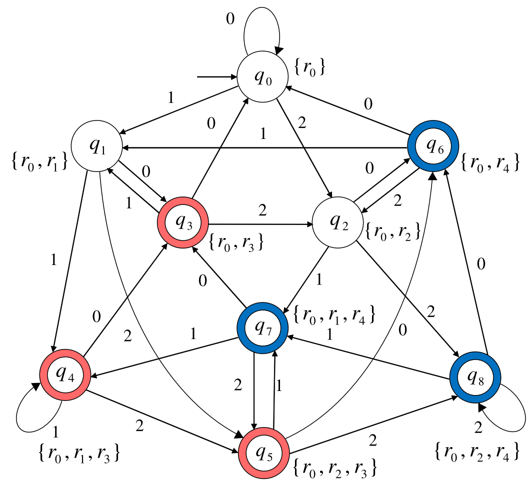

Fig. 7 DFA $D_{3,2}$ accepting $L_{3,2}$.

$L_{k,n}^{(i)} = \{x \in \{0, 1, \cdots, k-1\}^* \mid$ the nth symbol of x from its right end is i $\}$.

(Proof) The proof is a straightforward extension of binary case [13] to k-ary one. Suppose to the contrary that there is a DCFA M color-distinctly accepting $L_{k,n}$ whose number of states is less than k^n. Then, for two different strings of length n

$$x = a_1a_2 \cdots a_n,\ \ y = b_1b_2 \cdots b_n,$$

M will get into the same state, say q just after reading the right end symbols of them, because the number of different strings of length n over $\{0, 1, \ldots, k-1\}$ is k^n. Without loss of generality, for some $i = 1, \ldots, n$, it holds that

(1) $a_i = 0, b_i = j, j \ne 0,$

or

(2) $a_i = j_1, b_i = j_2, j_1 \ne j_2, j_1 \ne 0, j_2 \ne 0.$

Now, let

$$x' = x0^{i-1},\ \ y' = y0^{i-1}.$$

Since M is deterministic, it will get into the same state, say q' for both x' and y'. In the case of (1), from

$$x' \notin L_{k,n},\ \ y' \in L_{k,n},$$

M must get into a non-accepting state for x' and gets into an accepting state of some color for y'. In the case of (2), from

$$x' \in L_{k,n}^{(j_1)},\ \ y' \in L_{k,n}^{(j_2)}, j_1 \ne j_2,$$

M must get into accepting states of different colors for x' and y'. Both cases of (1) and (2) contradict the assumption. □

Fortunately, the colored finite automaton $N_{k,n}$ defined in the beginning of Sect. 4 was unmixed. On the other hands, M_0 arbitrarily made in Example 1 was mixed as shown in

Example 3. In Sect. 6, we consider the designing problems of unmixed NCFAs.

In order to claim that Theorem 2 is a language characterization of de Bruijn graph, we need a more rigid connection between automata and languages. That is, all state-minimized colored deterministic finite automata accepting a certain distinctly colored language are isomorphic. This uniqueness of state minimized DCFA can be proved in the same way as the case of noncolored ordinary DFA and is shown in Appendix.

5. Complexity Problems Concerning NCFA

In this section, we investigate computational complexities [19], [28], [29] of some decision problems concerning the unmixedness of nondeterministic colored finite automaton NCFA.

Definition 5: *Unmixedness verification* problem of nondeterministic colored finite automaton (abbreviated UV) is defined as follows.

$$\begin{cases} \text{Instance :} & \text{An NCFA } M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^{k} F_i), \\ \text{Question :} & \bigcup_{i=1}^{k} L_i(M) = \sum_{i=1}^{k} L_i(M)? \end{cases}$$

Theorem 3: The problem UV can be computed in polynomial time.

(Proof) We first show that under logarithmic cost criterion the complement $\overline{\text{UV}}$ of the problem is in the nondeterministic logarithmic space complexity class *NLOG*. Note that

$$\begin{aligned} &\langle M \rangle \in \overline{\text{UV}} \\ &\quad \Leftrightarrow \cup_{i=1}^{k} L_i(M) \neq \Sigma_{i=1}^{k} L_i(M) \\ &\quad \Leftrightarrow L_{i_1}(M) \cap L_{i_2}(M) \neq \emptyset \text{ for some } i_1 \neq i_2 \\ &\quad \Leftrightarrow x \in L_{i_1}(M), x \in L_{i_2}(M) \text{ for some } x \in \Sigma^*, i_1 \neq i_2 \\ &\quad \Leftrightarrow q_{f_1}, q_{f_2} \in \hat{\delta}(q_0, x) \text{ for some} \\ &\qquad q_{f_1} \in F_{i_1}, q_{f_2} \in F_{i_2}, x \in \Sigma^*, i_1 \neq i_2, \end{aligned}$$

where $\langle M \rangle$ denotes an appropriate coding of NCFA M. Given $\langle M \rangle$ of M, a log space-bounded Turing machine M' places two markers at the initial state q_0. Then, while guessing an input string $x \in \Sigma^*$, M' nondeterministically selects two adjacent states in accordance with the transition function δ and moves both markers to these next states. When M finishes reading x, M' enters an accepting state only if the states q_1 and q_2 where the two markers are placed satisfies

$$q_1 \in F_{i_1}, q_2 \in F_{i_2}, i_1 \neq i_2.$$

From

$$\overline{\text{UV}} \in NLOG,$$

we have

$$\text{UV} \in \text{co}-NLOG \subseteq \text{co}-P = P.$$

Next, we investigate under uniform cost criterion the practical complexity of UV problem. It is obvious that an instance of UV problem for an NCFA $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^{k} F_i)$ is equivalent to the instance of emptiness problem whether or not $L(M') = \emptyset$ for the direct product automaton $M' = (Q \times Q, \Sigma, \delta', (q_0, q_0), F')$ of M itself, where

$$\begin{aligned} &\delta'((p, q), a) \\ &\quad = \delta(p, a) \times \delta(q, a) \text{ for each } (p, q) \in Q \times Q, a \in \Sigma, \\ &F' = \{(p, q) \in Q \times Q \mid p \in F_{i_1}, q \in F_{i_2}, i_1 \neq i_2, \\ &\qquad i_1, i_2 \in \{1, 2, \ldots, k\}\}. \end{aligned}$$

Clearly, any instance of UV problem of size N can be deterministically transformed to the instance of the emptiness problem of size $O(N^2)$. The emptiness question of M' is to test whether no accepting state in F' can be reached from the initial state (q_0, q_0). This task can be done by using ordinary linear-time graph search algorithm. □

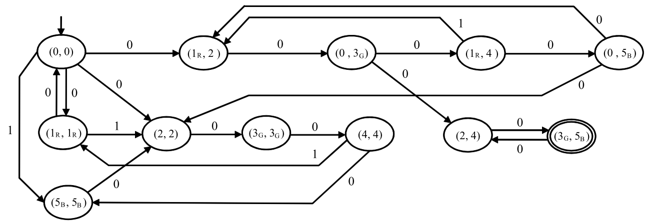

Fig. 8 The direct product automaton M_0' of M_0.

Example 5: Figure 8 shows the state transition diagram of the direct product automaton M_0' of M_0 defined in Example 1. ■

Corollary 1: The problem UV is *NLOG*-complete.

(Proof) It is known [28], [29] that *NLOG* is closed under complementation. From this and the first part of the proof of Theorem 4, we have

$$\text{UV} = \overline{\overline{\text{UV}}} \in \text{co}-NLOG = NLOG.$$

We next show that UV is *NLOG*-hard. We can point out that any NFA emptiness problem instance $L(M) = \emptyset$? is reducible to a $\overline{\text{UV}}$ problem instance $\langle M' \rangle$ as follows. Without loss of generality, we assume that M has one and only one accepting state q_f. The only difference of M' from M is that q_f is changed to a non-accepting state q_n and two different colored accepting states q_{f_1} and q_{f_2} are added both directly reachable from q_n by reading the same input symbol, say $a \in \Sigma$. In short,

$$\begin{aligned} L(M) \neq \emptyset &\Leftrightarrow \exists x \in \Sigma^*[x \in L(M)] \\ &\Leftrightarrow \exists x \in \Sigma^*[q_f \in \delta(q_0, x)] \\ &\Leftrightarrow \exists x \in \Sigma^*[q_{f_1}, q_{f_2} \in \delta'(q_0', xa)] \\ &\Leftrightarrow \exists x \in \Sigma^*[|I(xa)| = 2] \\ &\Leftrightarrow M' \text{ is mixed} \\ &\Leftrightarrow \langle M' \rangle \in \overline{\text{UV}}, \end{aligned}$$

where q_0, q'_0 are the initial states of M, M', respectively.

It is clear that the above modification of M to M' can be done by a deterministic logarithmic space-bounded Turing machine. That is,

$$\forall \mathrm{P} \in NLOG[\mathrm{P} \leq_{\log} \overline{UV}]$$

$$\Leftrightarrow \forall \overline{\mathrm{P}} \in \text{co}-NLOG = NLOG[\overline{\mathrm{P}} \leq_{\log} \overline{\overline{\mathrm{UV}}} = \mathrm{UV}]. \quad \square$$

Next, we investigate the potential capability of nondeterministic finite automata to be multi-colored and unmixed accepting machines.

Definition 6: *Unmixed partitioning* problem of nondeterministic finite automaton (abbreviated UP) is defined as follows.

$$\begin{cases} \text{Instance}: & \text{An NFA } M = (Q, \Sigma, \delta, q_0, F) \text{ and} \\ & \text{an integer } k \geq 2, \\ \text{Question}: & \text{Is there an unmixed NCFA } N = (Q, \Sigma, \delta, \\ & q_0, \sum_{i=1}^{k} F_i) \text{ such that } F = \sum_{i=1}^{k} F_i \text{ ?} \end{cases}$$

Fact 4: An NCFA $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^{k} F_i)$ is mixed

$$\Leftrightarrow \exists i, j[L_i(M) \cap L_j(M) \neq \emptyset]$$
$$\Leftrightarrow \exists i, j, x[x \in L_i(M) \cap L_j(M)]$$
$$\Leftrightarrow \exists i, j, x[\hat{\delta}(q_0, x) \cap F_i \neq \emptyset, \hat{\delta}(q_0, x) \cap F_j \neq \emptyset]$$
$$\Leftrightarrow \exists i, j, x, p, q[p \in F_i, q \in F_j, p, q \in \hat{\delta}(q_0, x)].$$

Definition 7: Let $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^{k} F_i)$ be an NCFA. The undirected graph $G = (V, E)$ obtained from the direct product automaton M' of M in the proof of Theorem 3 such that

$$\begin{cases} V = Q \\ E = \{(p, q) \in Q \times Q \mid \exists x \in \Sigma^*[(p, q) \in \hat{\delta}'((q_0, q_0), x)]\} \\ \quad = \{(p, q) \in Q \times Q \mid \exists x \in \Sigma^*[p, q \in \hat{\delta}(q_0, x)]\} \end{cases}$$

is called *simultaneously reachable graph* of M and denoted $G_{sr}(M)$.

Note that if M is deterministic, there is no edge in $G_{sr}(M)$. Simultaneously reachable graph of NFA will play crucial role in the following discussion. Obviously, the following holds.

Proposition 3: An NCFA M is mixed $\Leftrightarrow$ there exist $(p, q) \in E$ of $G_{sr}(M)$ such that $p \in F_i, q \in F_j$, for some $i \neq j$.

Example 6: Figure 9 shows the simultaneously reachable graph $G_{sr}(M_0)$ of M_0 defined in Example 1. There exists an edge $(3_G, 5_B)$ in the graph, where $3_G \in F_G, 5_B \in F_B$, which means that M_0 is mixed. ■

Lemma 1: An instance (M, k) of the UP problem is true if and only if the induced subgraph $G_{sr}(M)[F]$ of $G_{sr}(M)$ from the vertex subset F of Q, where F is the set of accepting states of M, has k or more connected components.

(Proof) If $G_{sr}(M)[F]$ has k or more connected components, then we can color the states of some k components among them with each different color. This never cause color collisions because two states in different components are never reached simultaneously from the initial state of M.

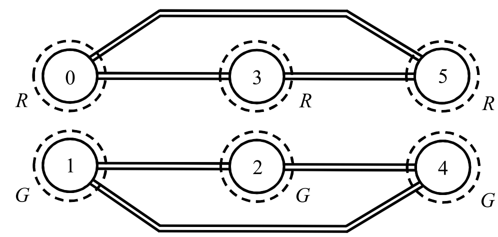

Fig. 9 The simultaneously reachable graph $G_{sr}(M_0)$ of M_0.

Conversely, if $G_{sr}[F](M)$ has less than k connected components, we cannot color them with k colors because we must color the all states of any component with the same color to avoid color collisions. □

Example 7: Let $M''_0 = (Q, \Sigma, \delta, q_0, F)$ be an ordinary NFA modified from M_0 in Example 3, where $Q = \{0, 1, 2, 3, 4, 5\}$ and $F = \{1, 3, 5\}$. As seen in Fig. 9, the UP instance $(M''_0, 3)$ is false since $G_{sr}(M''_0)[F]$ has just two connected components. ■

Theorem 4: The problem UP can be computed in polynomial time.

(Proof) As shown in the proof of Theorem 3, direct product automaton of an NCFA M can be constructed in polynomial time. By using a linear search of this automaton, we can get the simultaneously reachable graph $G_{sr}(M)$ and the induced graph $G_{sr}(M)[F]$. Connected components enumeration that adopts Lemma 1 can be done with an ordinary graph search algorithm in linear time. □

Next, we consider the problem whether we can select k unmixed state sets $F_1, \ldots, F_k$ from nonaccepting states of M other than the original accepting states set F_0. In this case, these k additional colors no more mean accepting situations but give us supplemental (e.g., error-related) informations of input when it halts in nonaccepting states.

Definition 8: *Unmixed extension* problem of nondeterministic finite automaton (abbreviated UE) is defined as follows.

$$\begin{cases} \text{Instance}: & \text{An NFA } M = (Q, \Sigma, \delta, q_0, F_0) \text{ and} \\ & \text{an integer } k \geq 1, \\ \text{Question}: & \text{Is there an unmixed NCFA } N = (Q, \Sigma, \delta, \\ & q_0, \sum_{i=0}^{k} F_i) \text{ ?} \end{cases}$$

Definition 9: For a graph $G = (V, E)$, if $\forall u, v \in I[(u, v) \notin E]$, then $I \subseteq V$ is called an independent set of G [8], [30], [31].

Definition 10: Independent set problem of undirected graph (abbreviated IS) is defined as follows.

$$\begin{cases} \text{Instance}: & \text{A graph } G \text{ and an integer } k \geq 2, \\ \text{Question}: & \text{Is there an independent set of size } k \text{ in } G \text{ ?} \end{cases}$$

Lemma 2: An instance (M, k) of the UE problem is true if and only if the instance $(G_{sr}(M)[\overline{F_0 \cup N(F_0)}], k)$ of the IS problem is true, where $G_{sr}(M)[\overline{F_0 \cup N(F_0)}]$ is the induced subgraph of $G_{sr}(M)$ from the complement set $Q - (F_0 \cup N(F_0))$ of the union of F_0 and its neighborhood vertex set $N(F_0)$ in $G_{sr}(M)$.

(Proof) Suppose the simultaneously reachable graph of M which the vertices of F_0 and their neighboring vertices $N(F_0)$ have been removed from the original $G_{sr}(M)$ has k independent vertices, which means that they are not direct neighbors with each other and also with the vertices in F_0. Thus, without color collisions we can color these k states with k different colors and the states in F_0 with one other color.

Conversely, when we select sets $F_1 + \cdots + F_k$ besides F_0 from the vertices of $G_{sr}(M)$, any $p \in F_i$ and any $q \in F_j$ must not be neighbors for $i \neq j$. Thus, $G_{sr}(M)$ which F_0 vertices and their neighbors are removed must have an independent set consisting of at least k elements. □

Theorem 5: The problem UE is NP-complete.

(Proof) We first show the NP-hardness of the problem by reducing the problem IS to this problem in polynomial time.

Let (G, k) be an instance of IS, where $G = (\{v_1, \cdots, v_n\}, E)$. We transform (G, k) to an instance $(M, k+1)$ of UE, where $M = (\{p_0, q_1, \cdots, q_n\}, \Sigma, \delta, p_0, \emptyset)$, $\Sigma = \{a_1, \cdots, a_n\} \cup \{a_{ij} \mid (i, j) \in I_E\}$, $I_E \triangleq \{(i, j) \mid (v_i, v_j) \in E\}$, $\delta(p_0, a_i) = \{q_i\}, i = 1, \ldots, n$, and $\delta(p_0, a_{ij}) = \{q_i, q_j\}, (i, j) \in I_E$.

It is clear that $G_{sr}(M)$ is the G added with the one isolated vertex which corresponds to the initial state p_0 of M and the size of input alphabet of M is bounded by $O(|G|)$, where $|G|$ is the size of G.

The nondeterministic polynomial-time solvability of UE follows from Lemma 2. □

It will be a future work to make the input alphabet of resulting NCFA of this polynomial reduction to be constant size, such as $\{0, 1\}$.

Example 8: Figure 10 shows an example of the transformation from a graph G which has an independent set of size 2 to its corresponding NFA M. Note that the simultaneously reachable graph $G_{sr}(M)$ of M is the same as G except that the initial state vertex is added. ■

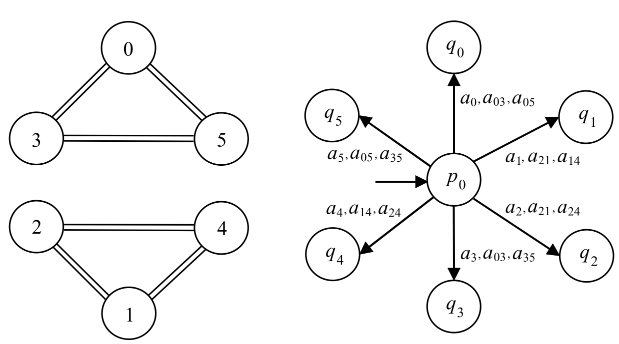

Fig. 10 The transformed NFA from a graph G.

6. Conclusion

In this paper, we first showed that general de Bruijn graph $DB_{k,n}$ is isomorphic to the minimum state deterministic colored finite automaton which accepts the colored language $\sum_{i=1}^{k-1} L_{k,n}^{(i)}$, where $L_{k,n}^{(i)}$ is the regular language of strings over $\{0, 1, \ldots, k-1\}$ whose ith symbols from the right ends are all i's.

We next investigated computational complexity problems concerning nondeterministic colored finite automata and showed some problems are solvable in polynomial time and another one is NP-complete. Simultaneously reachable graph introduced in this paper is inherent in any nondeterministic automaton not only in colored finite one and seems interesting in its own right to be investigated.

Of course, colored versions of conventional concepts such as regular expression or push-down automata remain to be investigated.

References

[1] G. Hoffmann de Visme, Binary Sequences, Hodder & Stoughton, 1971.
[2] H. Miyakawa, Y. Iwatare, and H. Imai, Coding Theory, Shoukoudou Co., 1973 (in Japanese).
[3] D. Du, F. Cao, and F. Hsu, "De Bruijn digraphs, kautz digraphs, and their generalizations," Combinatorial Network Theory, pp.65–96, Kluwer Academic, the Netherlands, 1996.
[4] T. Gagie, G. Manzini, and J. Sirén, "Wheeler graphs: A framework for BWT-based data structure," vol.698, pp.67–78, Theoretical Computer Science, 2017.
[5] Y. Tomata, Theory of Sequential Circuits, Shokodo, 1976 (in Japanese).
[6] De Bruijn graph, https://en.wikipedia.org/wiki/De-Bruijn-graphs
[7] C.S. Burris, F.C. Motta, and P.D. Shipman, "An Unoriented Variation on de Bruijn Sequences," Graphs and Combinatorics, vol.33, pp.845–858, 2017.
[8] A. Gibbons, Algorithmic Graph Theory, Cambridge University Press, 1985.
[9] M.N.S. Swamy and K. Thulasiraman, Graphs, Networks, and Algorithms, 1981.
[10] Z. Xu, Multi-Shift de Bruijn Sequence, Computer Science - Discrete Mathematics, 68R15, G.2.1, F.2.2, 2010.
[11] P.B. Dragon, O.I. Hernandez, J. Sawada, A. Williams, and D. Wong, "Constructing de Bruijn Sequences with Co-Lexicographic Order: The k-ary Grandmama Sequence," European Journal of Combinatorics, vol.72, pp.1–11, 2018.
[12] E. Moreno, "De Bruijn sequences and De Bruijn graphs for a general language," Information Processing Letters, vol.96, no.6, pp.214–219, 2005.
[13] J.E. Hopcroft, R. Motowani, and J.D. Ullman, "Intoroduction to Automaton Theory, Languages, and Computation I," Addison-Wesley Longman, 2001.
[14] J.E. Hopcroft and J.D. Ullman, "Introduction to Automata Theory, Languages, and Computation," Addison-Wesley Publishing, 1979.
[15] E. Moriya, Formal Language and Automaton, Saiensu-sha Co., Ltd. Publishers, 2001 (in Japanese).
[16] J.V. Leeuwen, Handbook of Theoretical Computer Science Volume B, The MIT Press/Elsevier, 1990.
[17] G. Castiglione, A. Restivo, and M. Sciortino, "Nondeterministic

Moore automata and Brzozowskis's, minimization algorithm," Theoretical Computer Science, vol.450, pp.81–91, 2012.
[18] J.R. Buchi, Finite Automata, Their Algebras and Grammars, Springer-Verlag New York, 1989.
[19] M.R. Garey and D.S. Johnson, Computers and Intractability A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, 1979.
[20] G. Rozenberg and A. Salomaa, "Handbook of Formal Languages," Volume 1 Word, Language, Grammar, Springer-Verlag, Berlin Heidelberg, 1997.
[21] L. Kallmeyer, Parsing Beyond Context-Free Grammars, Springer-Verlag Berlin Heidelberg, 2010.
[22] W. Kuich and H. Maurer, "The Structure Generating Function and Entropy of Tuple Languages," Information and Control, vol.19, no.3, pp.195–203, 1971.
[23] P. Linz, An Introduction to Formal Languages and Automata, Sixth Edition, Jones & Bartlett Learning, 2017.
[24] G. Jirásková and G. Pighizzini, "Optimal simulation of self-verifying automata by deterministic automata," Information and Computation, vol.209, no.3, pp.528–535, 2011.
[25] O. Kupferman and Y. Lustig, "Lattice Automata," Proceedings 8th International Conference Verification, Model Checking, and Abstract Interpretation, VMCAI 2007, Nice, France, pp.199–213, Jan. 2007,
[26] R.L. Graham, M. Grötschel, and L. Lovász, Handbook of Combinatorics volume 1, The MIT Press Cambridge, Massachusetts, 1995.
[27] S. Eilenberg, Automata, Languages, and Machines, Academic Press New York San Francisco London, 1976.
[28] M. Holzer and M. Kutrib, "Descriptional and computational complexity of finite automaton—A survey," Information and Computation, vol.209, no.3, pp.456–470, 2011.
[29] I.H. Sudborough, "On Tape-Bounded Complexity Classes and Multihead Finite Automata," Journal of Computer and System Sciences, vol.10, no.1, pp.62–76, 1975.
[30] L.W. Beineke and R.J. Wilson, Selected Topics in Graph Theory 2, Academic Press, 1983.
[31] F. Harary, Graph Theory, Addison-Wesley Publishing Company, 1972.
[32] A.L. Rosenberg, "The Pillars of Computation Theory," Springer Science + Business Media, 2010.

Appendix: Uniqueness of State Minimized DCFA

Here we give the proof of the uniqueness of deterministic colored finite automaton whose number of states is minimized.

Definition 11: Let $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^{k} F_i)$ be a DCFA. For any $i = 1, \cdots, k, q \in Q$, define

$$L_i(q) \triangleq \{x \in \Sigma^* \mid \hat{\delta}(q, x) \in F_i\},$$

i.e., the accepted language with ith color by M whose initial state is q instead of the originally given q_0.

Definition 12: Let $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^{k} F_i)$ be a DCFA. For any $p, q \in Q$, define

$$p \equiv q \Leftrightarrow L_i(p) = L_i(q) \ \text{ for each } \ i = 1, \ldots, k,$$

and say p and q are equivalent (or indistinguishable).

Fact 5: For a DCFA $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^{k} F_i)$,

$$\begin{aligned} p \equiv q \Leftrightarrow\ & (L_1(p), L_2(p), \ldots, L_k(p)) = \\ & (L_1(q), L_2(q), \ldots, L_k(q)) \\ \Leftrightarrow\ & (L_0(p), L_1(p), \ldots, L_k(p)) = \\ & (L_0(q), L_1(q), \ldots, L_k(q)), \end{aligned}$$

where $L_0(q) \triangleq \Sigma^* - \sum_{i=1}^{k} L_i(M)$. Therefore, from $\sum_{i=0}^{k} L_i(q) = \Sigma^*$,

$$\begin{aligned} p \not\equiv q \Leftrightarrow\ & L_i(p) \neq L_i(q), \text{ for some } \ i = 0, 1, \cdots, k \\ \Leftrightarrow\ & L_{i_1}(p) \neq L_{i_1}(q) \ \text{ and } \ L_{i_2}(p) \neq L_{i_2}(q), \\ & \text{for some } \ i_1 \neq i_2. \end{aligned}$$

□

In the following discussion, we use a natural extension of linear equation to represent NFA [20], [32].

Definition 13: Let $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^{k} F_i)$ be a DCFA. For any $q \in Q$, define the equation of state q:

$$q = a_1 q_1 + a_2 q_2 + \cdots + a_K q_K \, [+\varepsilon_i],$$

where $\delta(q, a_j) = q_j, j = 1, \cdots, K, K = |\Sigma|$ and the right most term $\varepsilon_i, i = 1, \ldots, k$ is added if and only if $q \in F_i$.

Fact 6: Let $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^{k})$ be a DCFA. The equation of $q \in Q$

$$q = a_1 q_1 + a_2 q_2 + \cdots + a_K q_K + \varepsilon_i$$

is equivalent to the language equation

$$L_j(q) = a_1 L_j(q_1) + a_2 L_j(q_2) + \cdots + a_K L_j(q_K),$$

for $j = 1, \ldots, k, j \neq i$ and

$$L_i(q) = a_1 L_i(q_1) + a_2 L_i(q_2) + \cdots + a_K L_i(q_K) + \varepsilon.$$

Theorem 6: (Uniqueness of minimum states DCFA) Given a distinctly colored language $\sum L_i$, the transition diagram of any minimum state DCFA which color-distinctly accepts $\sum L_i$ is isomorphic up to change of names, i.e., there exists a bijection between each pair of states of such two DCFAs.

(Proof) Suppose DCFAs $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^{k} F_i)$ and $M' = (Q', \Sigma, \delta', p_0, \sum_{i=1}^{k} F'_i)$ both accept $\sum_{i=1}^{k} L_i$. From $L(M) = L(M')$, it follows that $L_i(q_0) = L_i(p_0)$ for each $i = 1, \ldots, k$, thus, $q_0 \equiv p_0$. Let the equations of q_0 and p_0 be

$$\begin{cases} q_0 = a_1 q_1 + a_2 q_2 + \cdots + a_K q_K \, [+\varepsilon] \\ p_0 = a_1 p_1 + a_2 p_2 + \cdots + a_K p_K \, [+\varepsilon], \end{cases}$$

which are equivalent to

$$\begin{cases} L_i(q_0) = a_1 L_i(q_1) + \cdots + a_K L_i(q_K) \, [+\varepsilon], \\ \qquad i = 1, \ldots, K \\ L_i(p_0) = a_1 L_i(p_1) + \cdots + a_K L_i(p_K) \, [+\varepsilon], \\ \qquad i = 1, \ldots, K \end{cases}$$

$$\Leftrightarrow L_i(q_j) = L_i(p_j) \text{ for each } i = 1, \ldots, k, j = 1, \ldots, K$$
$$\Leftrightarrow q_j \equiv p_j \text{ for each } j = 1, \ldots, K.$$

In the same way, from the equations of q_j and p_j, we get the next equivalences between some states of M and M'. Continuing this process, we can finally conclude that any $q \in Q$ reachable from $q_0 \in Q$ is equivalent to some $p \in Q'$ reachable from $p_0 \in Q'$ and vice versa.

On the other hand, if there exist $q_1, q_2 \in Q, p \in Q', q_1 \neq q_2$ such that $q_1 \equiv p, q_2 \equiv p$, then we have $q_1 \equiv q_2$, which implies M is not minimum state DCFA. Thus, there must be no such state and the correspondence between Q and Q' must be one-to-one. □

Yoshiaki Takahashi received the B.E. and M.E. degrees in Faculty of Engineering, Yamaguchi University in 2006 and 2008, respectively. During 2011–2019, he stayed in Hofu Science Museum. He is now an assistant professor at the National Institute of Technology, Oshima College. Engaged in reserch on automata and language theory.

Akira Ito received the D.E. degrees in Faculty of Engineering, Nagoya University in 1992. He has been with Graduate School of Science and Technology for Innovation, Yamaguchi University since 1983 and is now an associate professor. Engaged in reserch on automata, formal language, algorithm, and complexity theory.