PAPER Special Section on Foundations of Computer Science — Foundations of Computer Science Supporting the Information Society —

On the Unmixedness Problems of Colored Pushdown Automata*

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

SUMMARY Recently, we introduced a new automata model, so-called colored finite automata (CFAs) whose accepting states are multi-colored (i.e., not conventional black-and-white acceptance) in order to classify their input strings into two or more languages, and solved the specific complexity problems concerning color-unmixedness of nondeterministic CFA. That is, so-called UV, UP, and UE problems are shown to be *NLOG*-complete, P, and *NP*-complete, respectively. In this paper, we apply the concept of colored accepting mechanism to pushdown automata and show that the corresponding versions of the above-mentioned complexity problems are all undecidable. We also investigate the case of unambiguous pushdown automata and show that one of the problems turns out to be permanent true (the others remain undecidable).
key words: *pushdown automata, undecidability, ambiguity*

1. Introduction

Finite Automaton is a fundamental and indispensable model of computation in numerous fields of modern computer science. Until now, a huge number of its variants have been proposed and investigated [1]–[9].

In the previous paper [10], we introduced a new generalized variant of finite automaton, called colored finite automaton (CFA), whose accepting states are multi-colored in order to classify their input strings into two or more languages. Although the original purpose of CFA was to find a specific finite automaton whose transition diagram is isomorphic to the generalized de Bruijn graph [11], it turns out that coloring of accepting states of nondeterministic automaton brings us a new perspective of nondeterministic computation.

Especially, when the coloring is not mixed i.e., the set of languages accepted with each color is mutually disjoint, each language accepted in the same color has conceptional coherency by itself and clear distinction to the others. Fortunately, the unmixedness of a given nondeterministic CFA can be checked efficiently, i.e., in a polynomial time. Therefore, CFA has practical potential to be used in applied areas of finite automaton. For example, colored regular expressions are introduced and their application to existing regular expression engines is proposed in [12].

In this paper, we apply this concept of accepting state coloring to the next level of Chomsky hierarchy, i.e., the context-free language class or equivalently the family of pushdown automata, and investigate the usefulness of colored pushdown automata and the computational complexity of their unmixedness problems. This is the answer to one of the open problems posed in [10].

This paper is organized as follows. Section 2 first recalls the definitions of CFA and gives a practical example of the coloring concept of finite automaton. We next give the definitions of colored pushdown automaton (CPDA) for the first time and gives an intriguing example of the coloring concept of pushdown automaton. Section 3 introduces the three computational problems concerning color-unmixedness of CPDA in the same way as CFA. These problems turn out to be undecidable at all, which is a sharp contrast with CFA cases. From this fact, Sect. 4 considers unambiguous pushdown automaton, i.e., a restricted automaton whose number of accepting paths is at most one. As a result, one of such restricted problems becomes trivially decidable and the others remain undecidable. Lastly, Sect. 5 summarizes the results and suggests a future direction of the research on CPDA.

2. Definitions and Examples

2.1 Colored Finite Automata

We first recall the definitions concerning colored finite automaton and give an example of the application of colored acceptance to ordinary finite automata.

Definition 1: [10] 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.

Definition 2: A *nondeterministic colored finite automaton* (with ε transition), abbreviated NCFA, is a 5-tuple $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^{k} F_i)$, where

Manuscript received April 9, 2022.
Manuscript revised August 30, 2022.
Manuscript publicized November 2, 2022.
†The author is with the National Institute of Technology, Oshima College, Yamaguchiken, 742–2193 Japan.
††The author is with the Graduate School of Science and Technology for Innovation, Yamaguchi University, Ube-shi, 755–8611 Japan.
*This manuscript is a revised version of the conference paper appeared in [13].
a) E-mail: takahashiy@oshima-k.ac.jp
b) E-mail: akito@yamaguchi-u.ac.jp (Corresponding author)
DOI: 10.1587/transinf.2022FCP0005

**For sets X and Y, direct sum X + Y is the union X ∪ Y satisfying the disjointness X ∩ Y = ∅. Thus, the direct sum $\sum_{i=1}^{k} L_i$ above is defined only if each L_i is mutually disjoint.

1. Q is a finite set of states,
2. Σ is a finite set of input symbols,
3. δ is the transition function from $Q \times (\Sigma \cup \{\varepsilon\})$ 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.

We denote by $\hat{\delta}(q, x)$ 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.

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

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.

We denote as ε-*Closure*(q) the set of states which are reachable from state p through ε transitions. For $p, q \in Q$, let $\varepsilon\text{-}Closure(p, q) \triangleq \{(p', q') \mid p' \in \varepsilon\text{-}Closure(p), q' \in \varepsilon\text{-}Closure(q)\}$.

Definition 3: Let $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^{k} F_i)$ be an NCFA. A 5-tuple $M' = (Q', \Sigma, \delta', q_0', F')$ is called direct product automaton of M itself, where

$$\begin{aligned} Q' &= \{\varepsilon\text{-}Closure(p, q) \mid (p, q) \in Q^2\}, \\ \delta'(\sigma, a) &= \{\varepsilon\text{-}Closure(p', q') \mid p' \in \delta(p, a), q' \in \delta(q, a), \\ &\qquad (p, q) \in \sigma\} \text{ for each } \sigma \in Q', a \in \Sigma, \\ q_0' &= \varepsilon\text{-}Closure(q_0, q_0), \text{ and} \\ F' &= \{\sigma \in Q' \mid (p, q) \in \sigma, p \in F_{i_1}, q \in F_{i_2}, i_1 \neq i_2, \\ &\qquad i_1, i_2 \in \{1, 2, \ldots, k\} \text{ for some } (p, q) \in \sigma\}. \end{aligned}$$

The definition of direct product automata above is slightly complicated than the original one [10] since we allow ε-transitions for colored finite automaton in Definition 2.

It is easily seen that $|Q'| \leq |Q|^2$ and M' is constructed from M in polynomial time.

Definition 4: 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 such that

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

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

† $X \triangleq Y$ means that X is defined as Y.
†† The same concept appears also in [14].

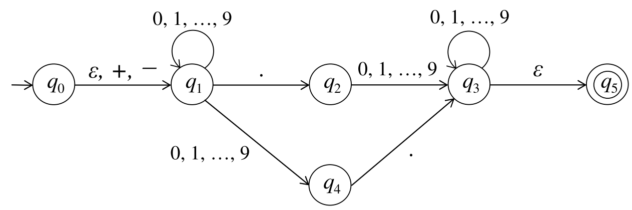

Fig. 1 ε−NFA M_{d} accepting decimal numbers [2]

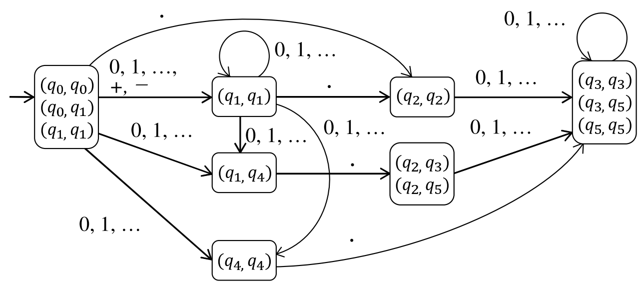

Fig. 2 The direct product automaton M_{d}' of Fig. 1

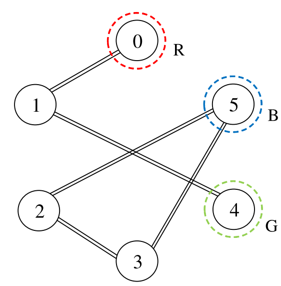

Fig. 3 The simultaneously reachable graph $G_{sr}(M_{\mathrm{d}})$ of Fig. 1

Example 1: Fig. 1 is an example of ordinary (non-colored) nondeterministic finite automaton M_{d} appeared in a familiar textbook [2]. M_{d} accepts decimal numbers each of which (1) may have positive or negative sign, (2) may not have integer part or fractional part of digits but must have either part, and (3) must have a decimal point. After constructing the product automaton M_{d}' from M_{d} as shown in Fig. 2, we get the simultaneously reachable graph $G_{sr}(M_{\mathrm{d}})$ of M_{d} as shown in Fig. 3.

From Fig. 3, we can select three independent vertices v_0, v_4, and v_5 which correspond to the states q_0, q_4, and q_5 of M_{d}, respectively. If we color them with three different colors R, G, and B, respectively, then the corresponding languages $L_{\mathrm{R}}(M_{\mathrm{d}})$, $L_{\mathrm{G}}(M_{\mathrm{d}})$, and $L_{\mathrm{B}}(M_{\mathrm{d}})$ are unmixed each other, where $L_{\mathrm{R}}(M_{\mathrm{d}}) = \{\varepsilon\}$, $L_{\mathrm{G}}(M_{\mathrm{d}})$ is the set of integer numbers having at least one digits without decimal points, and $L_{\mathrm{B}}(M_{\mathrm{d}}) = L(M_{\mathrm{d}})$.

Definition 5: [10] *Unmixedness verification* problem of nondeterministic colored finite automaton (abbreviated UV) is defined as follows.

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

Definition 6: [10] *Unmixed partitioning* problem of nondeterministic finite automaton (abbreviated UP) is defined as follows.

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

Definition 7: [10] *Unmixed extension* problem of nondeterministic finite automaton (abbreviated UE) is defined as follows.

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

In the case of NCFA, the computational complexities of the above three decision problems are *NLOG*-complete, P, and *NP*-complete, respectively [10].

2.2 Colored Pushdown Automata

The following is the definition of colored pushdown automaton that is a natural colored extension of usual pushdown automaton, abbreviated PDA [1]–[4], [6], [15].

Definition 8: A *nondeterministic colored pushdown automaton*, abbreviated NCPDA, is a 6-tuple $M=(Q,\Sigma,\Gamma,\delta, q_0,\sum_{i=1}^{k}F_i)$, where $\sum_{i=1}^{k}F_i\subseteq Q$ is a set of colored accepting states and F_i is a set of accepting states with ith color. The others are the same as normal NPDA.

In addition, a computational configuration (q,w,γ), a transition relation $\vdash$ between computational configurations, and its transitive closure $\vdash^*$ are defined in the same way as normal NPDA. $L_i(M)\triangleq\{x\in\Sigma^*\mid(q_0,x,\varepsilon)\vdash^*(q,\varepsilon,\varepsilon),q\in F_i\}$ is called the language accepted by M with ith color. Note that there exist two-fold conditions for M to accept a given input string such that (1) it must be in an accepting state and (2) the stack must be empty. The case of acceptance with only the condition (1), so-called *acceptance by final state*, will be described in the end of Sect. 4. $L(M)\triangleq\bigcup_{i=1}^{k}L_i(M)$ 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 the M is *color-distinctly* accepts $L(M)$.

Similarly, the CPDA version of simultaneously reachable graph is defined as follows.

Definition 9: Let $M=(Q,\Sigma,\Gamma,\delta,q_0,\sum_{i=1}^{k}F_i)$ be an NCPDA. The undirected graph $G=(V,E)$ obtained from M such that

$$\begin{cases} \bullet\ V=Q \\ \bullet\ E=\{(p,q)\in Q\times Q\mid\exists x\in\Sigma^*[(q_0,x,\varepsilon)\vdash^*(p,\varepsilon,\varepsilon) \\ \qquad \text{and } (q_0,x,\varepsilon)\vdash^*(q,\varepsilon,\varepsilon)]\} \end{cases}$$

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

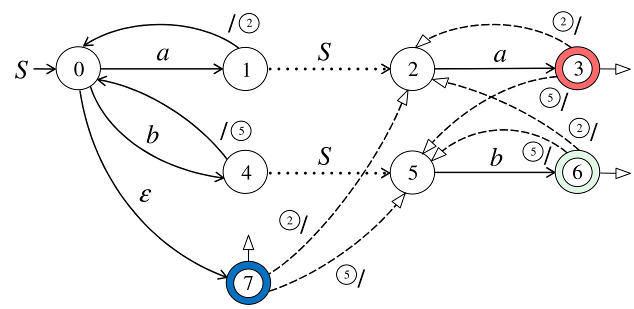

Fig. 4 The grammatical transition diagram of NCPDA M_{p}

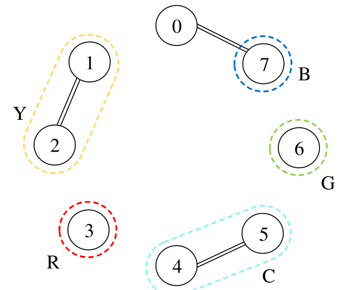

Fig. 5 The simultaneously reachable graph $G_{sr}(M_{\mathrm{p}})$ of NCPDA M_{p}

Example 2: Fig. 4 shows the grammatical transition diagram of NCPDA $M_{\mathrm{p}}=(\{q_0,q_1,\dots,q_7\},\{a,b\},\{\acute{q_2},\acute{q_5}\},\delta, q_0,\{q_3\}+\{q_6\}+\{q_7\})$, which is a multi-colored NPDA corresponding to the (unambiguous) grammar $S\to aSa\mid bSb\mid\varepsilon$ that generates the language $L_{\mathrm{pal}}=\{ww^{\mathrm{R}}\mid w\in\{a,b\}^*\}$ of palindromes. In the figure, "/ⓘ" and "ⓘ/" denote push and pop operations of stack symbol $\acute{q_i}$, respectively.

By letting the languages accepted in the accepting states q_3,q_6, and q_7 be denoted $L_{\mathrm{R}}(M_{\mathrm{p}}),L_{\mathrm{G}}(M_{\mathrm{p}})$, and $L_{\mathrm{B}}(M_{\mathrm{p}})$, respectively, it holds that

$$\begin{aligned} &L_{\mathrm{R}}(M_{\mathrm{p}})=aL_{\mathrm{pal}}a, \\ &L_{\mathrm{G}}(M_{\mathrm{p}})=bL_{\mathrm{pal}}b, \\ &L_{\mathrm{B}}(M_{\mathrm{p}})=\{\varepsilon\}, \text{ and} \\ &L(M_{\mathrm{p}})=\Sigma_{i=\mathrm{R,G,B}}L_i(M_{\mathrm{p}})=L_{\mathrm{pal}}, \end{aligned}$$

so M_{p} is color-distinctly accepts L_{pal}.

By further inquiries, it is seen that the individual languages accepted in the formerly nonaccepting states q_0,q_1,q_2,q_4, and q_5 are $L_0(M_{\mathrm{p}})=\{\varepsilon\}$, $L_1(M_{\mathrm{p}})=\{a\}$, $L_2(M_{\mathrm{p}})=aL_{\mathrm{pal}}$, $L_4(M_{\mathrm{p}})=\{b\}$, and $L_5(M_{\mathrm{p}})=bL_{\mathrm{pal}}$, respectively. From these identifications, we get the simultaneously reachable graph $G_{sr}(M_{\mathrm{p}})$ of M_{p} as shown in Fig. 5. If M_{p} undesirably halts in q_1 or q_2 which are Y-colored in the figure, we can see that symbol a is missing at the end of input string. Similarly, the case of halting in q_4 or q_5 which are C-colored in the figure tells us the missing of b.

3. The Unmixedness Problems of NCPDAs

In the same way as CFA, the UV, UP, and UE problems of NCPDA are defined as follows. Note that instance of the

UV problem is an NCPDA, while instances of the UP and UE problems are NPDAs.

Definition 10: *Unmixedness verification* problem of NCPDA (abbreviated UV) is defined as follows.

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

Definition 11: *Unmixed partitioning* problem of NPDA (abbreviated UP) is defined as follows.

$$\begin{cases} \text{Instance}: & \text{NPDA } M = (Q, \Sigma, \Gamma, \delta, q_0, F) \text{ and} \\ & \text{a positive integer } k, \\ \text{Question}: & \text{Is there an unmixed NCPDA } N = (Q, \Sigma, \Gamma, \delta, \\ & q_0, \sum_{i=1}^{k} F_i) \text{ such that } F = \sum_{i=1}^{k} F_i? \end{cases}$$

Definition 12: *Unmixed extension* problem of NPDA (abbreviated UE) is defined as follows.

$$\begin{cases} \text{Instance}: & \text{An NPDA } M = (Q, \Sigma, \Gamma, \delta, q_0, F_0) \text{ and} \\ & \text{a non-negative integer } k, \\ \text{Question}: & \text{Is there an unmixed NCPDA } N = (Q, \Sigma, \Gamma, \delta, \\ & q_0, \sum_{i=0}^{k} F_i)? \end{cases}$$

Theorem 1: (1) UV, (2) UP ($k = 2$), and (3) UE ($k = 1$) problems of NCPDA are all undecidable.

(**Proof**) We reduce the emptiness problem of Turing machines, which is known to be undecidable, to each problem. Let M be a Turing machine and Σ, Γ, p_0, F denote the set of input symbols, tape alphabet, initial state, and accepting states of M, respectively. The set

$$L\langle M\rangle = \{w_1\#{w_2}^{\mathrm{R}}\#w_3\#{w_4}^{\mathrm{R}}\#\cdots\#w_m(\text{or } {w_m}^{\mathrm{R}})\# \mid \\ w_1 \in p_0\Sigma^*, w_m \in \Gamma^* F \Gamma^*, w_i \vdash_M w_{i+1} \\ (1 \le i \le m-1)\}$$

of the valid accepting configuration sequences of a Turing machine M is expressed by the intersection of the languages

$$L(M_1) = \{w_1\#{w_2}^{\mathrm{R}}\#\cdots\#w_m(\text{or } {w_m}^{\mathrm{R}})\# \mid \\ w_1 \in p_0\Sigma^*, w_m \in \Gamma^* F \Gamma^*, w_{2i-1} \vdash_M w_{2i} \\ (1 \le i \le \lfloor m/2 \rfloor)\}$$

and

$$L(M_2) = \{w_1\#{w_2}^{\mathrm{R}}\#\cdots\#w_m(\text{or } {w_m}^{\mathrm{R}})\# \mid \\ w_1 \in p_0\Sigma^*, w_m \in \Gamma^* F \Gamma^*, w_{2i} \vdash_M w_{2i+1} \\ (1 \le i \le \lceil m/2 \rceil - 1)\}$$

accepted by the two deterministic PDAs M_1 and M_2, respectively, and it holds that $L(M) = \{x \in \Sigma^* \mid p_0x = w_1, w_1\#{w_2}^{\mathrm{R}}\#\cdots\#w_m(\text{or } {w_m}^{\mathrm{R}})\# \in L\langle M\rangle\} = \emptyset \Leftrightarrow L\langle M\rangle = \emptyset \Leftrightarrow L(M_1) \cap L(M_2) = \emptyset$ [3]. Note that if m is even (odd) then $2\lfloor m/2 \rfloor = m$ $(m-1)$ and $2(\lceil m/2 \rceil - 1) + 1 = m - 1$ (m).

An NPDA M' is constructed from the deterministic PDAs M_1 and M_2 as follows (see Fig. 6):

(i) we make a ε transition with a push operation of a new stack symbol Z_0 from the initial state q_i of M' to each initial states of M_1 and M_2, and

(ii) change the all accepting states of M_1 to nonaccepting states and make ε transitions with pop operations of Z_0 from them to a new accepting state q_{e1} of M'. In the same way, the lower M_2 part is modified to make ε transitions to another accepting state q_{e2}.

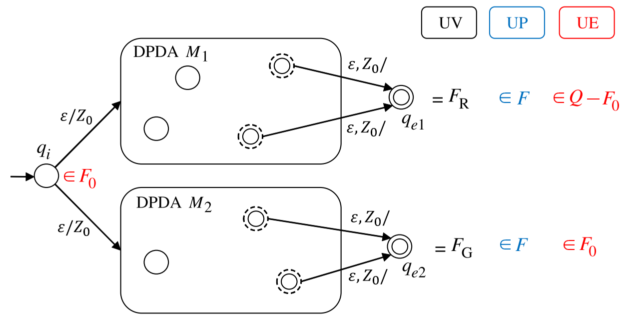

Fig. 6 A conceptual diagram of the proof of Theorem 1

Note that in the parts M_1 and M_2, no input can be regarded as accepted by M' due to the existence of Z_0 in the stack. From the construction above, we prove each undecidability as follows.

(1): For the two-colored NCPDA N which is changed from PDA M' with $F_{\mathrm{R}} = \{q_{e1}\}, F_{\mathrm{G}} = \{q_{e2}\}$, it holds that N is unmixed $\Leftrightarrow L_{\mathrm{R}}(N) \cap L_{\mathrm{G}}(N) = \emptyset \Leftrightarrow L(M) = \emptyset$.

(2): The claim that an unmixed NCPDA N can be obtained by splitting the set of accepting states $F = \{q_{e1}, q_{e2}\}$ of M' into two sets is true $\Leftrightarrow L_{\mathrm{R}}(N) \cap L_{\mathrm{G}}(N) = \emptyset \Leftrightarrow L(M) = \emptyset$.

(3): The NPDA M'' with $F_0 = \{q_i, q_{e2}\}$ and $q_{e1} \in Q - F_0$ can be changed to a two-colored unmixed PDA N' by adding another colored accepting set $F_1 = \{q_{e1}\} \Leftrightarrow L_0(N') \cap L_1(N') = (\{\varepsilon\} \cup L_{\mathrm{G}}(N)) \cap L_{\mathrm{R}}(N) = L_{\mathrm{G}}(N) \cap L_{\mathrm{R}}(N) = \emptyset \Leftrightarrow L(M) = \emptyset$. □

4. The Unmixedness Problems of UnAmbiguous NCPDAs

As shown in the previous section, unlike the case of colored finite automata, all the unmixedness problems are undecidable. In this section, we investigate the computational complexity of the problems of constrained NCPDA and NPDA, which are the inputs to the unmixedness problems.

Definition 13: We say that an NPDA is *unambiguous* if the following holds. If there exist accepting paths $P_1 \neq P_2$ such that $P_1 : (q_0, x, \varepsilon) \vdash^* (q_{f_1}, \varepsilon, \varepsilon)$ and $P_2 : (q_0, y, \varepsilon) \vdash^* (q_{f_2}, \varepsilon, \varepsilon)$, for some $q_{f_1}, q_{f_2} \in F$, then $x \neq y$.

Fact 1: [15] The transformation methods [1]–[3] used to show the equivalence of context-free grammars and NPDAs preserve their unambiguity.

We propose the two different concepts of ambiguity of NCPDA as follows.

Definition 14: (1) We say that an NCPDA is *unambiguous in the strong sense* if the following holds. If there exist accepting paths $P_1 \neq P_2$ such that $P_1 : (q_0, x, \varepsilon) \vdash^* (q_{f_1}, \varepsilon, \varepsilon)$ and $P_2 : (q_0, y, \varepsilon) \vdash^* (q_{f_2}, \varepsilon, \varepsilon)$, for some $q_{f_1}, q_{f_2} \in \bigcup_{i=0}^{k} F_i$, then $x \neq y$.

(2) We say that an NCPDA is *unambiguous in the weak sense* if the following holds. If there exist accepting paths $P_1 \neq P_2$ such that $P_1 : (q_0, x, \varepsilon) \vdash^* (q_{f_1}, \varepsilon, \varepsilon)$ and $P_2 : (q_0, y, \varepsilon) \vdash^* (q_{f_2}, \varepsilon, \varepsilon)$, for some $q_{f_1}, q_{f_2} \in F_i$ and for some $i (1 \leq i \leq k)$, then $x \neq y$.

The unambiguity in the strong sense of NCPDA is equal to the normal unambiguity when it is regarded as NPDA, ignoring color difference between its accepting states. On the other hand, in the case of the unambiguity in the weak sense, there may be two or more accepting paths (with different colors) when regarded as a normal NPDA.

Theorem 2: (1) UP instance $(M, |F|)$ for any unambiguous NPDA M is always true.
(2) UE problem of unambiguous NPDA is undecidable.
(3) UV problem of unambiguous NCPDA is (3-1) undecidable in the weak sense and (3-2) always true in the strong sense.

(**Proof**) (1): Suppose contrarily that some UP instance $(M, |F|)$ is false, i.e., an NCPDA N converted from M with $\Sigma_{i=1}^{k} F_i = F, F_i = \{q_{f_i}\}$ and $F = \{q_{f_1}, \ldots, q_{f_k}\}$ is mixed, which means that for some $i_1, i_2 (1 \leq i_1, i_2 \leq k), i_1 \neq i_2$, $L_{i_1}(N) \cap L_{i_2}(N) \neq \emptyset$, i.e., for some input x, there are two distinct accepting paths $P_1 : (q_0, x, \varepsilon) \vdash^* (q_{f_1}, \varepsilon, \varepsilon)$ and $P_2 : (q_0, x, \varepsilon) \vdash^* (q_{f_2}, \varepsilon, \varepsilon)$ of M on x, which contradicts the unambiguity of M.

(2): The NPDA M'' used in the proof of the part (3) of Theorem 1 to show the undecidability of the UE problems is natively unambiguous. This is because the nondeterministic transitions of M'' come out only at q_i, but the transition to the M_1 side never reaches the accepting state q_{e2}.

(3-1): For the NCPDA N used in the proof of the part (1) of Theorem 1 to show the undecidability of the UV problem, there is exactly one accepting path $(q_i, x, \varepsilon) \vdash^* (q_{e1}, \varepsilon, \varepsilon)$ for each $x \in L_{\mathrm{R}}(N)$ and exactly one accepting path $(q_i, y, \varepsilon) \vdash^* (q_{e2}, \varepsilon, \varepsilon)$ for each $x \in L_{\mathrm{G}}(N)$. Thus, N is unambiguous in the weak sense.

(3-2): Suppose contrarily that some NCPDA N which is unambiguous in the strong sense is mixed, which means that for some colors $i_1, i_2 (1 \leq i_1, i_2 \leq k), i_1 \neq i_2$, $L_{i_1}(N) \cap L_{i_2}(N) \neq \emptyset$, i.e., for some input x, there are two different accepting paths of N on x, which means that N taken as a non-colored NPDA is ambiguous. □

As announced in Sect. 2, we consider here the case of acceptance by final state only. The proofs of the theorems influenced by this change of accepting condition are those of UE problems (the part (3) of Theorem 1 and the part (2) of Theorem 2). Since we must conclude that a PDA accepts an input string even when it enters an accepting state without empty stack, one can yield as many colored languages as possible (at most the number of internal states of M_1 and M_2) if we leave the construction of NPDA M'' in the proof of Theorem 1 as it is.

To prevent such a situation, we modify the construction of M'' to the following.

(i) we change the two ε transitions with push operations of Z_0 from the initial state q_i to pure ε transitions (without stack operations), and
(ii) change the transitions with pop operations of Z_0 from the formerly accepting states of M_1 and M_2 to '\$'-reading transitions (without stack operations),

where a newly introduced symbol $\$ \notin \Sigma$ makes the languages accepted at states q_{e1} and q_{e2} do not intersect with the languages accepted at the other states. As elements of the set F_0 of original accepting states of M'', we select all the states in M_1 and M_2 in addition to q_{e2} and q_i. From these settings, we have q_{e1} as the only possible element of the set F_1 of extendable another-colored accepting states.

The set F_0 constructed in this way cannot be used for the proof of part (2) of Theorem 2 because the NPDA having such an F_0 as its accepting states may be ambiguous, i.e., there may exist two accepting paths on the same input string from the initial state q_i to some two accepting states in M_1 and M_2, respectively. To prevent such an ambiguity, we exclude the states in M_1 part from F_0 which are classified to be simultaneously reachable with states in M_2. Those simultaneously reachable states cannot be selected as members of F_1 because it causes a mixedness of the two colors. For states in M_1 which are classified not to be simultaneously reachable with states in M_2, we must have them remain in F_0. Even if there exist states in M_1 which cannot be determined whether or not they are simultaneously reachable with states in M_2 but we know that they behave the same way as q_{e1}, then we can conclude that the part (2) of Theorem 2 holds in the case of final state acceptance.

5. Summary and Discussions

Table 1 summarizes the results obtained in this paper, where "U" and "T" stand for undecidability and trivial decidability (permanent truth), respectively. Compared to colored finite automata, unconstrained colored PDAs exhibit the extreme difficulty (general impossibility) to check if they are unmixed or make them unmixed ones. On the other hand, when they are known as unambiguous ones, different situations emerge as seen in the second row. In fact, the unmixedness of the colored PDA M_{p} in Example 2 is not an accident but a direct consequence of its unambiguity. It is well-known, however, that the ambiguity itself is undecidable. Thus, we need to impose a stronger restriction upon PDAs, such as visibly (also called input-driven) PDA [16], [17], to make the unmixedness problems of colored PDAs have feasible solutions.

Table 1 decidabilities of the unmixedness problems of NCPDA

N(C)PDA	UV	UP	UE
general	U	U	U
unambiguous	weak: U strong: T	T	U

References

[1] M. Sipser, "Introduction to the Theory of Computation," PWS Publishing Company, 1997.
[2] J.E. Hopcroft, R. Motowani, and J.D. Ullman, "Introduction to Automaton Theory, Languages, and Computation," 3rd Edition, Addison-Wesley Longman, 2007.
[3] J.E. Hopcroft and J.D. Ullman, "Introduction to Automata Theory, Languages, and Computation," Addison-Wesley Publishing, 1979.
[4] J.V. Leeuwen, "Handbook of Theoretical Computer Science Volume B," The MIT Press/Elsevier, 1990.
[5] J.R. Buchi, "Finite Automata, Their Algebras and Grammars," Springer-Verlag New York, 1989.
[6] G. Rozenberg and A. Salomaa, "Handbook of Formal Languages Volume 1 Word, Language, Grammar," Springer-Verlag, Berlin Heidelberg, 1997.
[7] P. Linz, "An Introduction to Formal Languages and Automata," Sixth Edition, Jones & Bartlett Learning, 2017.
[8] S. Eilenberg, "Automata, Languages, and Machines," Academic Press New York San Francisco London, 1976.
[9] 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.
[10] Y. Takahashi and A. Ito, "Finite Automata with Colored Accepting States and Their Unmixedness Problems," Special section of IEICE Transactions on Information and Systems, vol.E105-D, no.3, pp.491–502, 2022.
[11] D. Du, F. Cao, and F. Hsu, "De Bruijn digraphs, kautz digraphs, and their generalizations," Combinatorial Network Theory, vol.1, pp.65–96, Kluwer Academic, the Netherlands, 1996.
[12] Y. Takahashi and A. Ito, "An Algorithm to Transform Colored Finite Automata to Colored Regular Expressions," to appear in The 2022 Joint Convention of Chugoku Section of Institutes of Electrical and Information Engineers, Japan, 2022 (in Japanese).
[13] Y. Takahashi and A. Ito, "On the Unmixedness Problems of Colored Pushdown Automata," Proceedings of NICS'21 8th NAFOSTED Conference on Information and Computer Science, pp.55–59, 2021.
[14] E. Indzhev and S. Kiefer, "On Complementing Unambiguous Automata and Graphs With Many Cliques and Cocliques," arXiv:2105.07470 [cs.FL], May 2021.
[15] C. Herzog, "Pushdown Automata with Bounded Nondeterminism and Bounded Ambiguity," Theoretical Computer Science, vol.181, no.1, pp.141–157, 1997.
[16] R. Alur and P. Madhusudan, "Visibly pushdown languages," Proceedings of the thirty-sixth annual ACM symposium on Theory of computing, pp.202–211, 2004.
[17] A. Okhotin and K. Salomaa, "Complexity of Input-Driven Pushdown Automata," ACM SIGACT News, vol.45, no.2, pp.47–67, 2014.

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

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 a senior lecturer at the National Institute of Technology, Oshima College. Engaged in research on automata and language theory.