A Two-Dimensional Extension of Insertion Systems

Kaoru Fujioka

International College of Arts and Sciences
Fukuoka Women's University
1-1-1 Kasumigaoka Higashi-ku, Fukuoka 813-8529, Japan
kaoru@fwu.ac.jp

Abstract. Insertion systems are extended to two-dimensional models that are used to generate picture languages. Insertion rules are defined in terms of rows and columns. Using picture-insertion rules, we herein introduce two types of derivations that depend on the position at which the rules are applied. We obtain the relationships between the classes of languages generated by picture-insertion systems for each type of derivation and a number of two-dimensional computing models, such as tiling systems. Furthermore, we introduce regular control for the derivations in picture-insertion systems. Finally, we compare the classes of languages generated by picture-insertion systems with and without regular control.

Keywords: picture languages, insertion systems, tiling systems.

1 Introduction

A number of approaches to represent and generate picture languages (two-dimensional languages), such as tiling systems, automata, regular expressions, and grammars [11], [5], [9], [1], have been reported. Some of the ideas behind these approaches are based on concepts related to string languages, and the corresponding results are proved which are extended from language properties into two-dimensional languages.

On the other hand, insertion and deletion systems are computing models that are based on the field of molecular biology and can characterize any recursively enumerable language that was originally defined for string languages.

Several methods for generating two-dimensional languages based on insertion and deletion operations have been proposed [8]. In [2], an array single-contextual insertion deletion system for two-dimensional pictures (ASInsDelP) was introduced based on DNA computation. With an insertion rule consisting of context-checking picture u and inserting picture x, a picture $\alpha uxu\beta$ is obtained from a given picture $\alpha u\beta$ by replicative transposition operation, which implies that columns are inserted by the insertion rule.

Computing models based on DNA molecules have evolved and increasingly complex structures have been introduced. Winfree [13] introduced a tile assembly model with DNA tile over two-dimensional arrays. A specialized model for DNA

A.-H. Dediu et al. (Eds.): TPNC 2014, LNCS 8890, pp. 181–192, 2014.

pattern assembly was proposed in [6] and theory of DNA pattern assembly has been recently developed.

In this paper, we focus on insertion operations with double context-checking strings while extending insertion systems from one dimension to two dimensions and then introduce picture-insertion systems to generate picture languages. The picture-insertion operation introduced herein is related to the (one-dimensional) insertion operations of the form (u, x, v) to produce a string $\alpha uxv\beta$ from a given string $\alpha uv\beta$ with context uv by inserting a string x between u and v [10].

A derivation proceeds using picture-insertion rules in order to generate arrays. In one step of the derivation, pictures of the same size are inserted in either every row or every column. We introduce two modes of applying picture-insertion rules: alongside mode and independent mode. In the alongside mode, pictures are inserted in the same column (resp. row) for any row (resp. column). In the independent mode, there is no restriction as to the position of insertion regarding rows or columns.

Furthermore, we introduce regular control for the derivations and demonstrated that the proposed control properly increases the generative powers of picture-insertion systems.

2 Preliminaries

In this section, we introduce the notation and basic definitions used in this paper. The basics of formal language theory are available in [11] and [10].

For an alphabet Σ, a *picture* p is a two-dimensional rectangular array of elements of Σ. Σ^{**} (resp. Σ^*) is the set of all pictures (resp. strings) over Σ, including the empty picture (resp. empty string) λ. A *picture language* (resp. language) over Σ is a subset of Σ^{**} (resp. Σ^*).

For a picture $p \in \Sigma^{**}$, let $\ell_1(p)$ (resp. $\ell_2(p)$) be the number of rows (resp. columns) of p. For a picture p in Σ^{**}, $|p| = (m, n)$ denotes the *size* of the picture p with $m = \ell_1(p)$ and $n = \ell_2(p)$. In particular, for a string w in Σ^*, $|w|$ denotes the length of w. For a string $w = a_1 a_2 \cdots a_n$, w^T is a vertical string, such as $\begin{matrix} a_1 \\ \vdots \\ a_n \end{matrix}$. For a picture p with $|p| = (m, n)$, the transposition of p is a picture q with $|q| = (n, m)$ such that rows and columns of p are interchanged.

For any $h \leq m$ and $k \leq n$, $B_{h,k}(p)$ is the set of all sub-pictures of p of size (h, k).

For pictures p and q, the *row and column concatenations* are denoted by $p \ominus q$ and $p \oplus q$, respectively, which are defined if $\ell_2(p) = \ell_2(q)$ (resp. $\ell_1(p) = \ell_1(q)$) holds. For $k \geq 0$, $p^{k\ominus}$ (resp. $p^{k\oplus}$) is the vertical (horizontal) juxtaposition of k p's. For picture languages L_1 and L_2, $L_1 \ominus L_2$ (resp. $L_1 \oplus L_2$) consists of pictures p such that $p = p_1 \ominus p_2$ (resp. $p = p_1 \oplus p_2$) with $p_1 \in L_1$ and $p_2 \in L_2$.

Next, we present a number of two-dimensional computing models. A *tile* is a square picture of size $(2, 2)$. For a finite set θ of tiles over alphabet $\Gamma \cup \{\#\}$, $LOC(\theta)$ denotes the set $\{p \in \Gamma^{**} \mid B_{2,2}(\hat{p}) \subseteq \theta\}$, where $\hat{p}$ is a picture obtained by surrounding p with the symbol #. A picture language L over Γ is *local* if $L = LOC(\theta)$ for some tile set θ.

For alphabets Γ and Σ, a coding $\varphi : \Gamma^* \to \Sigma^*$ is a morphism such that for any a in Γ, $\varphi(a) \in \Sigma$ holds. A projection $\pi : \Gamma^* \to \Sigma^*$ with $\Gamma \supseteq \Sigma$ is a morphism such that if a is in Σ then $\pi(a) = a$, otherwise $\pi(a) = \lambda$.

A *tiling system* is a tuple $\mathcal{T} = (\Sigma, \Gamma, \theta, \pi)$, where Σ and Γ are alphabets, θ is a finite set of tiles over the alphabet $\Gamma \cup \{\#\}$, and $\pi : \Gamma \to \Sigma$ is a projection. A language $L(\mathcal{T})$ defined by a tiling system $\mathcal{T}$ is $L(\mathcal{T}) = \pi(LOC(\theta))$. Let REC be the class of picture languages generated by tiling systems.

Based on a pure context-free rule of the form $a \to \alpha$ with $a \in \Sigma$ and $\alpha \in \Sigma^*$ in one dimension, a *pure 2D context-free grammar* $G = (\Sigma, P_c, P_r, A)$ is considered in [12] , where Σ is an alphabet, $P_c = \{t_{c_i} \mid 1 \leq i \leq m\}$, $P_r = \{t_{r_j} \mid 1 \leq j \leq n\}$, and $A \subseteq \Sigma^{**} - \{\lambda\}$ is a finite set of pictures over Σ. A *column table* t_{c_i} $(1 \leq i \leq m)$ is a set of pure context-free rules such that for any two rules $a \to \alpha$, $b \to \beta$ in t_{c_i}, $|\alpha| = |\beta|$ holds. Similarly, a *row table* t_{r_j} $(1 \leq j \leq n)$ is a set of pure context-free rules of the form $a \to \alpha^T$, $a \in \Sigma$, $\alpha \in \Sigma^*$ such that for any two rules $a \to \alpha^T$, $b \to \beta^T$ in t_{r_j}, $|\alpha| = |\beta|$ holds.

Let $P2DCFL$ be the class of picture languages generated by pure 2D context-free grammars.

A *context-free matrix grammar* (G_1, G_2) consists of two grammars G_1 and G_2, where

- $G_1 = (H_1, I_1, P_1, S)$ is a context-free grammar, H_1 is a finite set of horizontal nonterminals, $I_1 = \{S_1, \cdots, S_k\}$ is a finite set of intermediate symbols with $H_1 \cap I_1 = \emptyset$, P_1 is a finite set of context-free rules, S is the start symbol in H_1,
- $G_2 = (G_{21}, \cdots, G_{2k})$, where $G_{2i} = (V_{2i}, \Sigma, P_{2i}, S_{2i})$ with $1 \leq i \leq k$ is a regular grammar, V_{2i} is a finite set of nonterminals with $V_{2i} \cap V_{2j} = \emptyset$ for $i \neq j$, Σ is an alphabet, P_{2i} is a finite set of regular rules of the form $X \to aY$ or $X \to a$ with $X, Y \in V_{2i}$, $a \in \Sigma$, S_{2i} in V_{2i} is the start symbol.

A *regular matrix grammar* is a context-free matrix grammar (G_1, G_2), where both G_1 and G_2 are regular grammars. Let $CFML$ (resp. RML) be the class of picture languages generated by context-free (resp. regular) matrix grammars.

Let us conclude this section by presenting an insertion system for string languages [10], based on which we introduce a picture-insertion system in the next section. An *insertion system* is a tuple $\gamma = (\Sigma, P, A)$, where Σ is an alphabet, P is a finite set of *insertion rules* of the form (u, x, v) with $u, x, v \in \Sigma^*$, and A is a finite set of strings over Σ called axioms.

We write $\alpha \Longrightarrow \beta$ if $\alpha = \alpha_1 uv\alpha_2$ and $\beta = \alpha_1 uxv\alpha_2$ for some insertion rule $(u, x, v) \in P$ with $\alpha_1, \alpha_2 \in \Sigma^*$. The reflexive and transitive closure of $\Longrightarrow$ is defined as $\Longrightarrow^*$. A language generated by γ is defined as $L(\gamma) = \{w \in \Sigma^* \mid s \Longrightarrow^* w,$ for some $s \in A\}$.

Let INS be the class of string languages generated by insertion systems.

3 Picture-Insertion Systems

We introduce a *picture-insertion system* with two types of tables consisting of insertion rules for columns and rows, as follows:

Definition 1. *A* picture-insertion system is a tuple $\gamma = (\Sigma, I_c, I_r, A)$, where Σ is an alphabet, $I_c = \{t_{c_i} \mid 1 \le i \le m\}$, (resp. $I_r = \{t_{r_j} \mid 1 \le j \le n\}$) is a finite set of column (resp. row) tables, and A is a finite set of pictures over Σ.

Each t_{c_i} $(1 \le i \le m)$ is a set of C-type picture-insertion rules of the form (u, w, v) with $u, v \in \Sigma^*$ and $w \in \Sigma^+$ such that for any two rules (u, w, v) and (x, z, y) in t_{c_i}, we have $|u| = |x|$, $|w| = |z|$, and $|v| = |y|$.

Each t_{r_j} $(1 \le j \le n)$ is a set of R-type picture-insertion rules of the form $\begin{pmatrix} u, \\ w, \\ v \end{pmatrix}$ with $u^T, v^T \in \Sigma^*$ and $w^T \in \Sigma^+$ such that for any two rules $\begin{pmatrix} u, \\ w, \\ v \end{pmatrix}$ and $\begin{pmatrix} x, \\ z, \\ y \end{pmatrix}$ in t_{r_j}, we have $|u| = |x|$, $|w| = |z|$, and $|v| = |y|$.

Intuitively, a C-type (resp. R-type) rule refers to an insertion rule for a row (resp. column), then widen the column (resp. row) of pictures.

Next, we define two methods for applying insertion rules for pictures in order to obtain arrays.

Definition 2. *For pictures p_1 and p_2 in Σ^{**}, we say that p_1* derives *p_2 denoted by $p_1 \Longrightarrow_a p_2$ with* alongside mode *if p_2 is obtained from p_1 by inserting pictures with the same column (resp. row) for each row (resp. column) using C-type (resp. R-type) insertion rules of some t_{c_i} (resp. t_{r_j}) in I_c (resp. I_r).*

In a graphical representation of C-type picture-insertion rules, we have

$$\begin{array}{|c|c|c|c|} \hline \alpha_1 & u_1 & v_1 & \beta_1 \\ \cdots & \cdots & \cdots & \cdots \\ \alpha_{k'} & u_{k'} & v_{k'} & \beta_{k'} \\ \cdots & \cdots & \cdots & \cdots \\ \alpha_k & u_k & v_k & \beta_k \\ \hline \end{array} \Longrightarrow_a \begin{array}{|c|c|c|c|c|} \hline \alpha_1 & u_1 & w_1 & v_1 & \beta_1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \alpha_{k'} & u_{k'} & w_{k'} & v_{k'} & \beta_{k'} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \alpha_k & u_k & w_k & v_k & \beta_k \\ \hline \end{array}$$

We note that different C-type (resp. R-type) insertion rules might be applied for rows (resp. columns) in the process of $p_1 \Longrightarrow_a p_2$.

Definition 3. *For pictures p_1 and p_2 in Σ^{**}, we say that p_1 derives p_2 denoted by $p_1 \Longrightarrow_i p_2$ with the* independent mode *if p_2 is obtained from p_1 by inserting pictures for each row (resp. column) using C-type (reps. R-type) insertion rules of some t_{c_i} (resp. t_{r_j}) in I_c (resp. I_r).*

In a graphical representation of C-type picture-insertion rules, we have

$$\begin{array}{|c|c|c|c|}\hline \alpha_1 & u_1 & v_1 & \beta_1 \\ \hline \cdots & \cdots & \cdots & \cdots \\ \hline \alpha_{k'} & u_{k'} & v_{k'} & \beta_{k'} \\ \hline \cdots & \cdots & \cdots & \cdots \\ \hline \alpha_k & u_k & v_k & \beta_k \\ \hline \end{array} \Longrightarrow_i \begin{array}{|c|c|c|c|c|}\hline \alpha_1 & u_1 & w_1 & v_1 & \beta_1 \\ \hline \cdots & \cdots & \cdots & \cdots & \cdots \\ \hline \alpha_{k'} & u_{k'} & w_{k'} & v_{k'} & \beta_{k'} \\ \hline \cdots & \cdots & \cdots & \cdots & \cdots \\ \hline \alpha_k & u_k & w_k & v_k & \beta_k \\ \hline \end{array}$$

Different C-type (resp. R-type) insertion rules might be applied to rows (resp. columns) in the process of $p_1 \Longrightarrow_i p_2$.

Unlike the alongside mode in Definition 2*, there is no restriction regarding the position at which to apply picture-insertion rules in the process of* $p_1 \Longrightarrow_i p_2$.

The reflexive and transitive closure of $\Longrightarrow_a$ (resp. $\Longrightarrow_i$) is defined as $\Longrightarrow_a^*$ (resp. $\Longrightarrow_i^*$). A picture language generated by $\gamma = (\Sigma, I_c, I_r, A)$ using the alongside mode (resp. independent mode) is defined as $L_a(\gamma) = \{w \in \Sigma^{**} \mid s \Longrightarrow_a^* w$, for some $s \in A\}$ (resp. $L_i(\gamma) = \{w \in \Sigma^{**} \mid s \Longrightarrow_i^* w$, for some $s \in A\}$).

Let $INPA$ (resp. $INPI$) be the class of picture languages generated by picture-insertion systems using the alongside mode (resp. independent mode).

4 Examples

In the following, we present examples of picture-insertion systems.

Example 4. Consider a picture-insertion system $\gamma_1 = (\Sigma, I_c, I_r, A)$, where $\Sigma = \{a, b\}$, $I_c = \{t_{c1}\}$ with $t_{c1} = \{(\lambda, ab, \lambda)\}$, $I_r = \{t_{r1}\}$ with $t_{r1} = \{\begin{pmatrix} a, \\ a, \\ \lambda \end{pmatrix}, \begin{pmatrix} b, \\ b, \\ \lambda \end{pmatrix}\}$, $A = \{\lambda\}$.

The picture language $L_a(\gamma_1)$ generated by γ_1 using the alongside mode is $\{w^{k\ominus} \mid w$ is Dyck's string language over $\{a, b\}$, $k \geq 0$ $\}$. For example, the following pictures are generated using the alongside mode:

$$ab,\ \begin{array}{c} ab \\ ab \end{array},\ \begin{array}{c} ab \\ ab \\ ab \end{array},\ \begin{array}{c} aabb \\ aabb \end{array},\ \begin{array}{c} aabb \\ aabb \\ aabb \end{array},\ \begin{array}{c} aabbab \\ aabbab \end{array},\ \begin{array}{c} aabbab \\ aabbab \\ aabbab \end{array},\ \begin{array}{c} aabbabab \\ aabbabab \\ aabbabab \\ aabbabab \end{array}.$$

From the R-type insertion rules in t_{r1} and the definition of the alongside mode, all of the row strings are the same.

On the other hand, we consider the picture language $L_i(\gamma_1)$ generated by γ_1 using the independent mode. For example, the following pictures are generated using the independent mode:

$$ab,\ \begin{array}{c} ab \\ ab \end{array},\ \begin{array}{c} ab \\ ab \\ ab \end{array},\ \begin{array}{c} aabb \\ abab \end{array},\ \begin{array}{c} aabb \\ abab \\ abab \end{array},\ \begin{array}{c} aaabbb \\ ababab \end{array},\ \begin{array}{c} aaabbb \\ ababab \\ ababab \end{array},\ \begin{array}{c} aaaabbbb \\ aaaabbbb \\ abababab \\ abababab \end{array}.$$

Any picture generated by γ_1 using the independent mode satisfies the condition that any row must consist of a Dyck language over Σ. From the definition of the independent mode, $L_i(\gamma_1)$ includes a picture with different row strings.

Furthermore, we have the inclusion $L_a(\gamma) \subset L_i(\gamma)$.

As shown in Example 4, picture-insertion systems are two-dimensional generalizations of insertion systems in linear cases. Actually, for the case with $P_r = \emptyset$, the picture-insertion system generates strings using both the alongside mode and the independent mode. Then, a Dyck language is generated, as noted in Lemma 7. Note that a Dyck language is not regular (in a one-dimensional sense), which implies that both $INPA$ and $INPI$ include a picture language which is not regular.

Example 5. Consider a picture-insertion system $\gamma_2 = (\Sigma, I_c, I_r, A)$, where

$\Sigma = \{a, b, d, e\}$,
$I_c = \{t_{c1}, t_{c2}\}$,
$I_r = \{t_{r1}, t_{r2}\}$,
$t_{c1} = \{(u, ab, \lambda) \mid u \in \{a, b\}\} \cup \{(u, de, \lambda) \mid u \in \{d, e\}\}$,
$t_{c2} = \{(\lambda, ab, v) \mid v \in \{a, b\}\} \cup \{(\lambda, de, v) \mid v \in \{d, e\}\}$,

$$t_{r1} = \{\begin{pmatrix} u, \\ w, \\ \lambda \end{pmatrix} \mid w = (ad)^T,\ u \in \{a, d\}\} \cup \{\begin{pmatrix} u, \\ w, \\ \lambda \end{pmatrix} \mid w = (de)^T,\ u \in \{d, e\}\},$$

$$t_{r2} = \{\begin{pmatrix} \lambda, \\ w, \\ v \end{pmatrix} \mid w = (be)^T,\ u \in \{b, e\}\} \cup \{\begin{pmatrix} \lambda, \\ w, \\ v \end{pmatrix} \mid w = (be)^T,\ v \in \{b, e\}\},$$

$$A = \{ \begin{matrix} ab \\ de \end{matrix} \}.$$

The following are examples of the pictures generated by γ using the alongside mode:

$$\begin{matrix} aabb \\ ddee \end{matrix},\ \begin{matrix} aaabbb \\ dddeee \end{matrix},\ \begin{matrix} abab \\ dede \end{matrix},\ \begin{matrix} ababab \\ dedede \end{matrix},\ \begin{matrix} aabb \\ aabb \\ ddee \\ ddee \end{matrix},\ \begin{matrix} abab \\ dede \\ abab \\ dede \end{matrix}.$$

Note that the pictures generated by γ_2 using the alongside mode are *Chinese boxes*, which are nested boxes with two-dimensional Dyck analogue structures. The symbol a (resp. b, d, and e) implies the upper left (resp. upper right, lower left, and lower right) corner of the box.

5 Properties and Comparisons of Picture-Insertion Systems Using the Alongside Mode

We first consider picture-insertion systems using the alongside mode and obtain the following result.

Lemma 6. *The class of $INPA$ is not closed under the operations of union, column catenation, or row catenation. The class is closed under transposition.*

Proof. Consider the picture language $L_a(\gamma_1)$ in Example 4 and the picture language L_1, which is obtained by replacing b with d in $L_a(\gamma_1)$.

Suppose that there is a picture-insertion system $\gamma' = (\{a, b, d\}, P_c', P_r', A')$ such that $L_a(\gamma_1) \cup L_1 = L_a(\gamma')$. For infinite pictures over $\{a, b\}$ in $L_a(\gamma_1)$, there is a picture-insertion rule (u, w, v) with $|w|_a = |w|_b$. Similarly, for L_1, we have a picture-insertion rule (x, z, y) with $|z|_a = |z|_d$. In order to generate only pictures in $L_a(\gamma_1) \cup L_2$, any picture-insertion rule (u, w, v) with $|w|_a = |w|_b$ satisfies $|uv|_b > 0$. Otherwise, γ' generates a picture p which satisfies $|p|_b > 0$ and $|p|_d > 0$.

Similarly, for infinite pictures over $\{a, d\}$ in L_1, a picture-insertion rule (x, z, y) with $|z|_a = |z|_d$ satisfies $|xy|_d > 0$.

Let $n = max\{|u|, |v|, |w| \mid (u, w, v)$ be a picture-insertion rule for $\gamma'\}$, $\ell_a = max\{\ell_1(\alpha) \mid \alpha \in A'\}$, and $N > n + \ell_a$. Consider the string $a^N aba^N b^{2N}$ in $L_1 \subset L_a(\gamma')$. There is no way to generate the string by a picture-insertion rule (u, w, v) with $|uv|_b > 0$ and $|w|_a = |w|_b$ due to the substring ab between a^N and a^N.

The non-closure property under column catenation and row catenation can be determined by considering $L_a(\gamma_1) \ominus L_2$ and $L_a(\gamma_1) \oplus L_2$, respectively.

The closure property under transposition can be determined if we substitute column (resp. row) tables with row (resp. column) tables by replacing the rule (u, w, v) (resp. $\begin{pmatrix} u^T, \\ w^T, \\ v^T \end{pmatrix}$) with $\begin{pmatrix} u^T, \\ w^T, \\ v^T \end{pmatrix}$ (resp. (u, w, v)) and consider transposition axiom. □

From the construction of tiling systems, defined by the projection of local languages, tiling systems are considered to be two-dimensional generalizations of one-dimensional regular languages. In one-dimensional cases, for the class of regular languages denoted by REG, the proper inclusion $REG \subset INS$ holds, where INS is the class of languages generated by insertion systems in one dimension. In contrast to the one-dimensional cases, for the class of picture languages generated by tiling systems denoted by REC, we obtain the following result.

Lemma 7. *The class of $INPA$ is incomparable with the class of REC.*

Proof. Consider a picture-insertion system $\gamma_2 = (\Sigma, I_c, I_r, A)$, where $\Sigma = \{a, b\}$, $I_c = \{t_{c1}\}$ with $t_{c1} = \{(\lambda, ab, \lambda)\}$, $I_r = \emptyset$, $A = \{\lambda\}$ derived from Example 4.

The class of REC coincides with that of regular languages if restricted to one dimension. A language $L_a(\gamma_2)$ in a one-dimensional language is a Dyck language which is not regular. Therefore, there is a picture-insertion system γ_2 such that $L_a(\gamma_2)$ is not generated by a tiling system.

Consider a picture language L_s over $\{a, b\}$, where L_s consists of squares, the positions in the main diagonal of which are covered by a and the remaining squares are covered by b. From [11], L_s is in the class of REC.

Suppose that there is a picture-insertion system γ such that $L_s = L_a(\gamma)$. For a picture w in $\{a, b\}^{**}$, there is a derivation $w \Longrightarrow_a w'$ using the C-type insertion rule such that $|w| = (m, m)$ and $|w'| = (m, m')$ with $m' > m$. For the picture w' in L_s, $|w'| = (m, m')$ with $m \neq m'$ holds. Thus, we have a contradiction.

Thus, the lemma is proved. □

We compare the class of $INPA$ to the class of $P2DCFL$ as follows.

Lemma 8. *The class of $INPA$ is incomparable with the class of $P2DCFL$.*

Proof. Consider a pure 2D context-free grammar $G = (\Sigma, P_c, P_r, \{ \begin{smallmatrix} aba \\ ded \end{smallmatrix} \})$, where $\Sigma = \{a, b, d, e\}$, $P_c = \{t_c\}$, $P_r = \{t_r\}$ with $t_c = \{b \to aba,\ e \to ded\}$ and $t_r = \{a \to \begin{smallmatrix} a \\ d \end{smallmatrix}, b \to \begin{smallmatrix} b \\ e \end{smallmatrix}\}$ [12].

Suppose that there is a picture-insertion system $\gamma = (\Sigma, I_c, I_r, A)$ such that $L(G) = L_a(\gamma)$. Any row in $L(G)$ consists of strings such that $a^n ba^n$ or $d^n ed^n$ with $n \geq 1$.

There is no picture-insertion rule that can generate these strings, which can be proved by contradiction. Briefly, a picture-insertion rule (u, w, v) with $w \in \{a\}^*$ (resp. $w \in \{d\}^*$) needed for infinitely long $a^n ba^n$ (resp. $d^n ed^n$) generates a string $a^i ba^j$ (resp. $d^i ed^j$) with $i \neq j$.

On the other hand, consider a picture-insertion system $\gamma_3 = (\{a, b\}, I_{c3}, I_{r3}, A_3)$ such that $I_{c3} = \{t_{c3}\}$ with $t_{c3} = \{(ab, ab, \lambda)\}$, $I_{r3} = \{t_{r3}\}$ with $t_{r3} = \{\begin{pmatrix} a, \\ a, \\ \lambda \end{pmatrix}, \begin{pmatrix} b, \\ b, \\ \lambda \end{pmatrix}\}$, $A_3 = \{a^3b^3ab, a^3b^3\}$. A picture language $L_a(\gamma_3)$ consists of pictures such that $(a^3b^3(ab)^n)^{m\ominus}$ with $m \geq 1$, $n \geq 0$. From [7][12], there is no pure 2D context-free grammar which generates $L_a(\gamma_3)$.

Thus, the lemma is proved. □

In the following, we consider two types of matrix grammars. First, from the picture language $L_a(\gamma_2)$ in Lemma 7 and the fact that a Dyck language is not regular in a one-dimensional sense, we obtain the following result.

Corollary 9. *There is a picture language in the class of $INPA$ which is not in the class of RML.*

Lemma 10. *Every picture language in the class of RML is a coding of a language in the class of $INPA$.*

Proof. (Outline)
The proof is based on the idea that in a one-dimensional sense, the class of regular languages REG is included in the class of insertion systems INS [10].

Consider a regular matrix grammar (G_1, G_2), where $G_1 = (H_1, I_1, P_1, S)$ and $G_2 = (G_{21}, \cdots, G_{2k})$ with $G_{2i} = (V_{2i}, \Sigma, P_{2i}, S_{2i})$ $(1 \leq i \leq k)$ are regular.

For regular languages $L(G_1)$ and $L(G_{2i})$ $(1 \leq i \leq k)$, there are picture-insertion systems $\gamma'_1 = (I_1, P_1, A_1)$, $\gamma'_{2i} = (\Sigma, P_{2i}, A_{2i})$ and integers n_1, n_{2i} such that $L(G_1) = L(\gamma'_1)$, $L(G_{2i}) = L(\gamma'_{2i})$, and the axiom in γ'_1 (resp. γ'_{2i}) is no more than $n_1 - 1$ (resp. $n_{2i} - 1$). (See [10] for more details about how to define the integers n_1 and n_{2i}.)

Let N be the least common multiple of n_{2i} $(1 \leq i \leq k)$.

We construct a picture-insertion system with the additional symbols $\gamma = (\Sigma \cup \{S_{2i} \mid 1 \leq i \leq k\} \cup \{\#\}, I_c, I_r, A)$ and a coding $\varphi : (\Sigma \cup \{S_{2i} \mid 1 \leq i \leq k\} \cup \{\#\})^* \to \Sigma^*$ with $\varphi(a) = a$ for $a \in \Sigma$ and $\varphi(a) = \lambda$ otherwise.

Roughly speaking, the regular language $L(G_1)$ is simulated by C-type picture-insertion rules in γ and the regular language $L(G_{2i})$ is simulated by R-type picture-insertion rules in γ. Finally, the coding φ deletes the redundant symbols S_{2i} (resp. $\#$) required to simulate G_{2i} (resp. G_1).

A finite set of pictures A satisfies $A = \{ \begin{matrix} w \\ \#^n \end{matrix} \mid w \in A_1,\ |w| = n\}$. We construct C-type picture-insertion rules (u, w, λ) and $(\#^m, \#^n, \lambda)$, where (u, w, λ) is in P_1 concerning γ'_1 and $|u| = m$, $|w| = n$. The symbol $\#$ lies in the bottommost of each picture.

We construct R-type picture-insertion rules

– R1-type: $\begin{pmatrix} S_{2i}, \\ z_{2i}, \\ \# \end{pmatrix}$, where $z_{2i} \in L(G_{2i})$, $\ell_1(z_{2i}) \leq N-1$

– R2-type: $\begin{pmatrix} u_{2i}, \\ w_{2i}, \\ \lambda \end{pmatrix}$, $\ell_1(u_{2i}) \leq N-1$, $1 \leq \ell_1(w_{2i}) \leq N$ derived from P_{2i}.

Let I_c (resp. I_r) consists of column (resp. row) tables, where each table includes all the C-type (resp. R-type) picture-insertion rules with the same length of triplet.

By the context-checking of picture-insertion rules, concerning R-type picture-insertion rules, the R1-type rules are applied first and only once. Then, R2-type picture-insertion rules are used to simulate $L(G_{2i})$.

The topmost row generated by G_1 can be simulated by C-type picture-insertion rules, and each column can be simulated by R-type picture-insertion rules. Finally, we eliminate the symbols S_{2i} and # using the coding φ. □

Corollary 11. *The class of INPA is incomparable with the class of CFML.*

Proof. In the one-dimensional case, the class of insertion systems is incomparable with that of context-free languages. Thus, the corollary holds for these one-dimensional language relationships. □

6 Properties and Comparisons of Picture-Insertion Systems Using the Independent Mode

Next, we consider picture-insertion systems using the independent mode and obtain the following results.

Lemma 12. *The class of INPI is not closed under the operations of union, column catenation, or row catenation. The class is closed under transposition.*

Proof. Consider the picture languages $L_i(\gamma_1)$ in Example 4 and the picture language L_4 which is obtained by placing d in the place of b as for $L_i(\gamma_1)$.

The proof is similar to the proof for Lemma 6 for picture-insertion systems using the alongside mode. □

For the generative powers, in the following, we compare picture-insertion systems with tiling systems.

Lemma 13. *The class of INPI is incomparable with the class of REC.*

Proof. The proof is similar to the proof for Lemma 7. □

Lemma 14. *The class of INPI is incomparable with the class of P2DCFL.*

Proof. The proof is almost the same as Lemma 8.

Consider a pure 2D context-free grammar $G = (\Sigma, P_c, P_r, \{ \begin{smallmatrix} aba \\ ded \end{smallmatrix} \})$ in Lemma 8. As in Lemma 8, we can prove by contradiction that there is no picture-insertion system γ such that $L(G) = L_i(\gamma)$.

Next, we consider a picture-insertion system $\gamma_3 = (\{a, b\}, I_c, I_r, A)$ in Lemma 8. A picture language $L_i(\gamma_3)$ consists of pictures such that $(a^3b^3(ab)^n)^{m\ominus}$ with $m \geq 1$ and $n \geq 0$.

From the above two results, we obtain the claim. □

Note 15. Consider a picture-insertion system $\gamma = (\Sigma, I_c, I_r, A)$, the C-type rule of which is of the form (a, w, λ) with $a \in \Sigma$, $w \in \Sigma^*$, and the R-type rule is of the form $\begin{pmatrix} a, \\ w^T, \\ \lambda \end{pmatrix}$ with $a \in \Sigma$, $w^T \in \Sigma^*$.

Then, there is a pure 2D context-free grammar G such that $L_i(\gamma) = L(G)$. The proof is obvious from the definition. For example, a C-type picture-insertion rule (a, w, λ) can be simulated by the rule $a \to aw$. Therefore, a restricted insertion system using the independent mode is simulated by a pure 2D context-free grammar.

Lemma 16. *Both, $INPA$ and $INPI$ are incomparable.*

Proof. Consider the picture-insertion system γ_1 over $\Sigma = \{a, b\}$ in Example 4. We show that the picture language $L_a(\gamma_1)$ in $INPA$ is not in $INPI$.

Suppose that there is a picture-insertion system γ such that $L_a(\gamma_1) = L_i(\gamma)$. Each column table in γ consists of one C-type picture-insertion rule. For any string w in a Dyck language, and C-type picture-insertion rule, the derivation in γ proceeds deterministically, i.e., there should be only one place where the picture-insertion rule can be applied. Otherwise, the picture-insertion rule can generate a picture with different row strings.

For a C-type picture-insertion rule (u, w, v) which satisfies $\alpha uv\beta \Longrightarrow_a \alpha uwv\beta$ with $\alpha, u, w, v, \beta \in \Sigma^*$ and $\alpha uv\beta, \alpha uwv\beta \in L_a(\gamma_1)$. For the string $\alpha uv\beta\alpha uv\beta$ in $L_a(\gamma_1)$, there are two substrings uv in $\alpha uv\beta\alpha uv\beta$ for which we can apply the picture-insertion rule (u, w, v). Thus, we have a contradiction.

On the other hand, we show that the picture language $L_i(\gamma_1)$ in $INPI$ is not in $INPA$. Suppose that there is a picture-insertion system $\gamma' = (\Sigma, I_c, I_r, A)$ such that $L_i(\gamma_1) = L_a(\gamma')$. Let $n = max\{\ell_1(\alpha) \mid \alpha \in A\} + max\{|uwv| \mid (u, w, v)$ be a C-type picture-insertion rule in $I_c\}$.

Consider a picture $\begin{smallmatrix} a^{2n}b^{2n}ab \\ aba^{2n}b^{2n} \end{smallmatrix}$. For the first row $a^{2n}b^{2n}ab$, a picture-insertion rule is applied to the nested structure of $a^{2n}b^{2n}$ at least twice. On the other hand, for the second row $aba^{2n}b^{2n}$, the substring ab without a nested structure is followed by $a^{2n}b^{2n}$. Therefore, there is no way to generate the picture using the alongside mode.

From the above two results, we obtain the claim. □

7 Picture-Insertion Systems with Regular Control

We introduce an additional function for picture-insertion systems to control the application of picture-insertion rules. As is noted in [12], controlling the application with a regular language remains the generative power in general. However, the control for pure 2D context-free grammars properly increases their generative power. We apply regular control to picture-insertion systems and present the results below.

Definition 17. *A picture-insertion system with regular control* $\gamma(R)$ *is a tuple* (γ, R)*, where* $\gamma = (\Sigma, I_c, I_r, A)$ *is a picture-insertion system and* R *is a regular language over* Γ *with a set of table labels of* I_c *and* I_r*.*

For pictures p_1 and p_2 in Σ^{**}, we write $p_1 \overset{t}{\Longrightarrow}_a p_2$ (resp. $p_1 \overset{t}{\Longrightarrow}_i p_2$) if p_1 derives p_2 using the alongside mode (resp. independent mode) using C-type (resp. R-type) insertion rules in the column table t in I_c (resp. I_r).

For a picture-insertion system $\gamma = (\Sigma, I_c, I_r, A)$ and a regular language R over Γ, a regular control picture-insertion language $L_a(\gamma(R))$ using the alongside mode is a set of pictures $w \in \Sigma^{**}$ such that $S \overset{\alpha}{\Longrightarrow}_a w$, $\alpha \in \Gamma^*$, $\alpha \in R$. Similarly, a regular control picture-insertion language $L_i(\gamma(R))$ using the independent mode is a set of pictures $w \in \Sigma^{**}$ such that $S \overset{\alpha}{\Longrightarrow}_i w$, $\alpha \in \Gamma^*$, $\alpha \in R$.

Let $INPAC$ (resp. $INPIC$) be a set of regular control picture-insertion languages using the alongside mode (resp. independent mode).

Example 18. Consider a picture-insertion system $\gamma_1 = (\Sigma, I_c, I_r, A)$ in Example 4 and a regular language $R = \{(t_{c1}t_{r1}t_{r1})^n \mid n \geq 1\}$ with $t_{c1} = \{(\lambda, ab, \lambda)\}$ and $t_{r1} = \{\begin{pmatrix} a, \\ a, \\ \lambda \end{pmatrix}, \begin{pmatrix} b, \\ b, \\ \lambda \end{pmatrix}\}$.

The C-type insertion rule in t_{c1} inserts picture ab and widens two columns. Each R-type rule in t_{r2} inserts the picture a or b and widens one row. The regular control language $(t_{c1}t_{r1}t_{r1})^n$ with $n \geq 1$ enables the generated pictures to be proportionate to the lengths of rows and columns.

Lemma 19. $INPA \subset INPAC$*.* $INPI \subset INPIC$*.*

Proof. From the definition of $INPAC$ and $INPIC$, the inclusions $INPA \subseteq INPAC$ and $INPI \subseteq INPIC$ are obvious.

As noted in Lemma 7, there is no picture-insertion system γ such that $L(\gamma)$ consists of squares.

From Example 18, there is a picture language which consists of infinitely many square pictures in $INPAC$ and $INPIC$. Then, the proper inclusion is proved. □

The lemmas imply that regular control properly increases generative power for picture-insertion systems.

8 Concluding Remarks

In this paper, we introduced picture-insertion systems which generate picture languages and for the language classes generated by picture-insertion systems, we considered comparisons with two-dimensional computing models. Furthermore, in order to perform the derivations, we defined regular control of the picture-insertion systems, which properly increases the generative powers.

In the future, as in the one-dimensional case, picture insertion-*deletion* systems can be defined in which we can use not only picture-insertion operations but also *deletion* operations.

Using insertion systems together with some morphisms, characterizing and representation theorems have been given for the one-dimensional case [3] [4]. We discuss whether similar representation theorems are possible in the two-dimensional case.

Acknowledgments. The present study was supported by JSPS KAKENHI Grant Number 23740081.

References

1. Cherubini, A., Crespi-Reghizzi, S., Pradella, M., Pietro, P.S.: Picture languages: Tiling systems versus tile rewriting grammars. Theoretical Computer Science 356(1-2), 90–103 (2006)
2. Easwarakumar, K.S., Murugan, A.: Possibilities of constructing two dimensional pictures in DNA computing: Part I. Int. J. Comput. Math. 82(11), 1307–1321 (2005)
3. Fujioka, K.: Morphic characterizations of languages in Chomsky hierarchy with insertion and locality. Inf. Comput. 209(3), 397–408 (2011)
4. Fujioka, K.: Morphic characterizations with insertion systems controlled by a context of length one. Theoretical Computer Science 469, 69–76 (2013)
5. Inoue, K., Nakamura, A.: Some properties of two-dimensional on-line tessellation acceptors. Information Science 13(2), 95–121 (1977)
6. Ma, X., Lombardi, F.: Synthesis of tile sets for DNA self-assembly. IEEE Trans. on CAD of Integrated Circuits and Systems 27(5), 963–967 (2008)
7. Maurer, H.A., Salomaa, A., Wood, D.: Pure grammars. Information and Control 44(1), 47–72 (1980)
8. Murugan, A., Easwarakumar, K.S.: Possibilities of constructing two dimensional pictures in DNA computing: Part II. Int. J. Comput. Math. 83(1), 1–20 (2006)
9. Pradella, M., Cherubini, A., Reghizzi, S.C.: A unifying approach to picture grammars. Information and Computation 209(9), 1246–1267 (2011)
10. Păun, G., Rozenberg, G., Salomaa, A.: DNA Computing. New Computing Paradigms. Springer (1998)
11. Rozenberg, G., Salomaa, A. (eds.): Handbook of formal languages. Springer-Verlag New York, Inc., New York (1997)
12. Subramanian, K.G., Ali, R.M., Geethalakshmi, M., Nagar, A.K.: Pure 2D picture grammars and languages. Discrete Appl. Math. 157(16), 3401–3411 (2009)
13. Winfree, E., Liu, F., Wenzler, L., Seeman, N.: Design and Self-assembly of Two-dimensional DNA Crystals. Nature 394, 539–544 (1998)