数理解析研究所講究録
第 1894 巻 2014 年 28-31

A note on picture insertion systems

Kaoru Fujioka *

1 Introduction

Insertion and deletion systems are computing models based on the field of molecular biology. Several proposals have been made for generating two-dimensional languages based on insertion and deletion with replicative transposition operation.

In this paper, we focus on insertion operations and extend an insertion system from one dimension (1D) to 2D then introduce a picture insertion system to generate picture languages. The picture insertion operation introduced in this paper relates to the insertion operations in one dimensions 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 [2]. We also present some examples and results concerning picture insertion systems.

2 Preliminaries

In this section, we introduce notation and basic definitions that are necessary for this paper. The basic notions and definitions in formal language theory are found in [4].

For an alphabet T, a *picture* p is a two-dimensional rectangular array of elements of T. T^{**} is the set of all pictures over T. A *picture language* over T is a subset of T^{**}.

For a picture $p \in T^{**}$, let $\ell_1(p)$ (resp. $\ell_2(p)$) be the number of rows (resp. columns) of p. For a picture p in T^{**}, $|p| = (m, n)$ denotes the *size* of the picture p with $m = \ell_1(p)$ and $n = \ell_2(p)$.

The *row and column concatenations* are denoted $p \ominus q$ and $p \oplus q$, respectively, and defined if p and q have the same number of columns (resp. lows). $p^{k\ominus}$ (resp. $p^{k\oplus}$) is the vertical (horizontal) juxtaposition of k's p.

A *tiling system* [3] 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 Γ, and $\pi : \Gamma \rightarrow \Sigma$ is a projection. Let TS be the class of picture languages generated by tiling systems.

3 Picture insertion systems

Definition 1 *A picture insertion system is a tuple $\gamma = (T, P, A)$, where T is an alphabet, P is a finite set of picture insertion rules, and A is a finite set of pictures over T. P may contain the following three types of picture insertion rules:*

- *R-type:* (u, w, v), *where* $\ell_1(u) = \ell_1(v) = \ell_1(w)$.
- *C-type:* $\begin{pmatrix} u, \\ w, \\ v \end{pmatrix}$, *where* $\ell_2(u) = \ell_2(v) = \ell_2(w)$.
- *RC-type:* $\begin{pmatrix} u, & w_1, & v, \\ w_2, & w_3, & w_4, \\ x, & w_5, & y \end{pmatrix}$, *where* $\ell_1(u) = \ell_1(w_1) = \ell_1(v)$, $\ell_1(w_2) = \ell_1(w_3) = \ell_1(w_4)$, $\ell_1(x) = \ell_1(w_5) = \ell_1(y)$, $\ell_2(u) = \ell_2(w_2) = \ell_2(x)$, $\ell_2(w_1) = \ell_2(w_3) = \ell_2(w_5)$, $\ell_2(v) = \ell_2(w_4) = \ell_2(y)$, *and* $w_3 \neq \lambda$.

*Department of Environmental Science, International College of Arts and Sciences, Fukuoka Women's University

Intuitively, R-type (resp. C-type) rule means an insertion rule in row (resp. column), that is, the picture w is inserted in between the pictures u and v. An RC-type rule is intend to insert the pictures w_i $(1 \leq i \leq 5)$ into the picture consisting of u,v,x, and y. We break up the rectangle into subpictures u,v,x, and y and secure the cross-shaped space, then insert those pictures.

We show how to apply insertion rules in the following definition.

Definition 2 *For pictures p_1, p_2 in T^{**}, we say that p_1 derives p_2 in one step if*

- *there is an R-type rule (u, w, v) with $u, v, w \in T^{m*}$ for $m \geq 1$ such that $p_1 = \alpha \obar u \obar v \obar \beta$ and $p_2 = \alpha \obar u \obar w \obar v \obar \beta$ with $\alpha, \beta \in T^{m*}$. We write $p_1 \rightharpoonup_R p_2$. In a graphical representation, it means*

$$\begin{array}{|c|c|c|c|}\hline \alpha & u & v & \beta \\ \hline \end{array} \rightharpoonup_R \begin{array}{|c|c|c|c|c|}\hline \alpha & u & w & v & \beta \\ \hline \end{array}$$

- *there is a C-Type rule $\begin{pmatrix} u, \\ w, \\ v \end{pmatrix}$ with $u, v, w \in T^{*n}$ for $n \geq 1$ such that $p_1 = \alpha \ominus u \ominus v \ominus \beta$ and $p_2 = \alpha \ominus u \ominus w \ominus v \ominus \beta$ with $\alpha, \beta \in T^{*n}$. We write $p_1 \rightharpoonup_C p_2$. In a graphical representation, it means*

$$\begin{array}{|c|}\hline \alpha \\ \hline u \\ \hline v \\ \hline \beta \\ \hline \end{array} \rightharpoonup_C \begin{array}{|c|}\hline \alpha \\ \hline u \\ \hline w \\ \hline v \\ \hline \beta \\ \hline \end{array}$$

- *there is an RC-Type rule $\begin{pmatrix} u, & w_1, & v \\ w_2, & w_3, & w_4 \\ x, & w_5, & y \end{pmatrix}$ such that $p_1 = (u \obar v) \ominus (x \obar y)$ and $p_2 = (u \obar w_1 \obar v) \ominus (w_2 \obar w_3 \obar w_4) \ominus (x \obar w_5 \obar y)$. We write $p_1 \rightharpoonup_{RC} p_2$. In a graphical representation, it means*

$$\begin{array}{|c|c|}\hline u & v \\ \hline x & y \\ \hline \end{array} \Longrightarrow \begin{array}{|c|c|c|}\hline u & w_1 & v \\ \hline w_2 & w_3 & w_4 \\ \hline x & w_5 & y \\ \hline \end{array}$$

If there is no confusion, we write $\rightharpoonup$ instead of $\rightharpoonup_R$, $\rightharpoonup_C$, and $\rightharpoonup_{RC}$. The reflexive and transitive closure of $\rightharpoonup$ (resp. $\rightharpoonup_R$, $\rightharpoonup_C$) is defined as $\rightharpoonup^*$ (resp. $\rightharpoonup_R^*$, $\rightharpoonup_C^*$). The transitive closure of $\rightharpoonup$ (resp. $\rightharpoonup_R$, $\rightharpoonup_C$) is denoted by $\rightharpoonup^+$ (resp. $\rightharpoonup_R^+$, $\rightharpoonup_C^+$).

With $\rightharpoonup_R$, $\rightharpoonup_C$, and $\rightharpoonup_{RC}$ we introduce a *standard derivation* denoted by $\Longrightarrow$ in the following definition.

Definition 3 *For pictures p_1 and p_2, $p_1 \Longrightarrow p_2$ is defined in the following three cases:*

1. [Using R-type rules]

- *pictures p_1 and p_2 satisfy $p_1 = (\alpha_1 \obar \beta_1) \ominus \cdots \ominus (\alpha_n \obar \beta_n)$ and $p_2 = (\alpha_1 \obar w_1 \obar \beta_1) \ominus \cdots \ominus (\alpha_n \obar w_n \obar \beta_n)$, where for each $1 \leq i \leq n$,*
  - *there is a derivation $\alpha_i \obar \beta_i \rightharpoonup_R^* \alpha_i \obar w_i \obar \beta_i$,*
  - *there are $l_a, l_b, l_w \geq 0$ such that $\ell_2(\alpha_i) = l_a$, $\ell_2(\beta_i) = l_b$, $\ell_2(w_i) = l_w$,*
- *there is no picture p' in T^{**} such that $p_1 \rightharpoonup_R^+ p' \rightharpoonup_R^+ p_2$.*

In a graphical representation, it means

$$\begin{array}{|c|c|}\hline \alpha_1 & \beta_1 \\ \hline \cdots & \cdots \\ \hline \alpha_n & \beta_n \\ \hline \end{array} \Longrightarrow \begin{array}{|c|c|c|}\hline \alpha_1 & w_1 & \beta_1 \\ \hline \cdots & \cdots & \cdots \\ \hline \alpha_n & w_n & \beta_n \\ \hline \end{array}$$

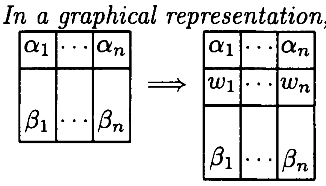
2. [Using C-type rules]

- *pictures p_1 and p_2 satisfy $p_1 = (\alpha_1 \ominus \beta_1) \obar \cdots \obar (\alpha_n \ominus \beta_n)$ and $p_2 = (\alpha_1 \ominus w_1 \ominus \beta_1) \obar \cdots \obar (\alpha_n \ominus w_n \ominus \beta_n)$, where for each $1 \leq i \leq n$,*

- *there is a derivation* $\alpha_i \ominus \beta_i \rightharpoonup_C^* \alpha_i \ominus w_i \ominus \beta_i$,
- *there are* $l_a, l_b, l_w \geq 0$ *such that* $\ell_1(\alpha_i) = l_a$, $\ell_1(\beta_i) = l_b$, $\ell_1(w_i) = l_w$,

• *there is no picture* p' *in* T^{**} *such that* $p_1 \rightharpoonup_C^+ p' \rightharpoonup_C^+ p_2$.

In a graphical representation, it means



3. [Using an RC-type rule]

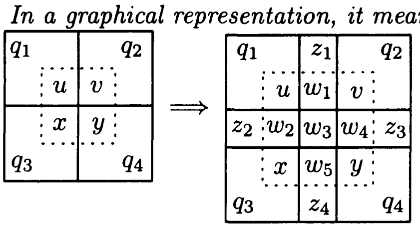
• *there is an RC-type rule* $\begin{pmatrix} u, & w_1, & v, \\ w_2, & w_3, & w_4, \\ x, & w_5, & y \end{pmatrix}$,

• *pictures* p_1 *and* p_2 *satisfy* $p_1 = (q_1 \oplus q_2) \ominus (q_3 \oplus q_4)$ *and* $p_2 = (q_1 \oplus (z_1 \ominus w_1) \oplus q_2) \ominus (z_2 \oplus x \oplus w_3 \oplus w_4 \oplus z_3) \ominus (q_3 \oplus (w_5 \ominus z_4) \oplus q_4)$,

• *the lower right corner (resp. lower left, upper right, upper left) of* q_1 *(resp.* q_2, q_3, q_4*) is* u *(resp.* v, x, y*),*

• z_1 *(resp.* z_2, z_3, z_4*) is inserted by R-type (resp. C-type, C-type, R-type) rules.*

In a graphical representation, it means



Intuitively, the standard derivation $\Longrightarrow$ is the smallest unit to applied to a picture by applying picture insertion rules. The reflexive and transitive closure of $\Longrightarrow$ is defined as $\Longrightarrow^*$.

A picture language generated by $\gamma = (T, P, A)$ is defined as $L(\gamma) = \{w \in T^{**} \mid s \Longrightarrow^* w$, for some $s \in A\}$.

A picture insertion system $\gamma = (T, P, A)$ is said to be of *weight* $(i, j; k, l)$ if the number of rows (resp. columns) for context checking picture is not more than i (resp. j), and the number of rows (resp. columns) for inserted picture is not more than k (resp. l).

For $i, j, k, l \geq 0$, let $INS_{k,l}^{i,j}$ be the class of picture languages generated by picture insertion systems of weight $(i', j'; k', l')$ with $i' \leq i$, $j' \leq j$, $k' \leq k$, and $l' \leq l$. If some of the parameters i, j, k, l are not bounded, we use $*$ in place of the symbols for those parameters.

Example 1 Consider a picture insertion system $\gamma = (T, P, A)$, where $T = \{a, b\}$, $P = \{(\lambda, ab, \lambda)\}$, $A = \{\lambda\}$. The picture language generated by γ is viewed as a Dyck's string language.

As shown in Example 1, picture insertion systems are 2D generalizations of insertion systems in 1D cases. We slightly note that Dyck language is not regular (in 1D sense).

Example 2 Consider a picture insertion system $\gamma = (T, P, A)$, where $T = \{a, b\}$, $P = \{(\lambda, \begin{array}{c} ab \\ ba \end{array}, \lambda), \begin{pmatrix} \lambda, \\ \begin{array}{c} ab \\ ba, \end{array} \\ \lambda \end{pmatrix}\}$, $A = \{\lambda\}$.

The followings are some of the pictures generated by γ; λ, $\begin{array}{c} ab \\ ba \end{array}$, $\begin{array}{c} abab \\ baba \end{array}$, $\begin{array}{c} aabb \\ bbaa \end{array}$, $\begin{array}{c} ab \\ ba \\ ab \\ ba \end{array}$, $\begin{array}{c} ab \\ ab \\ ba \\ ba \end{array}$, $\begin{array}{c} ababab \\ bababa \end{array}$, $\begin{array}{c} aaabbb \\ bbbaaa \end{array}$, $\begin{array}{c} aabbab \\ bbaaba \end{array}$.

For example, the picture $\begin{array}{c} aabbab \\ ccddcd \end{array}$ is derived in two ways as follows: $\lambda \Longrightarrow \begin{array}{c} ab \\ ba \end{array} \Longrightarrow \begin{array}{c} abab \\ baba \end{array} \Longrightarrow \begin{array}{c} aabbab \\ bbaaba \end{array}$, $\lambda \Longrightarrow \begin{array}{c} ab \\ ba \end{array} \Longrightarrow \begin{array}{c} aabb \\ bbaa \end{array} \Longrightarrow \begin{array}{c} aabbab \\ bbaaba \end{array}$.

Example 3 Consider a picture insertion system $\gamma = (T, P, A)$, where $T = \{a, b\}$, $P = \{(b, b, \lambda),$

$\begin{pmatrix} b, \\ b, \\ \lambda \end{pmatrix}, \begin{pmatrix} a, & b, & b \\ b, & a, & b \\ b, & b, & a \end{pmatrix} \}, A = \{ \begin{smallmatrix} ab \\ ba \end{smallmatrix} \}.$

A derivation in γ proceeds as follows:

$$\begin{smallmatrix} ab \\ ba \end{smallmatrix} \Longrightarrow \begin{smallmatrix} abb \\ bab \\ bba \end{smallmatrix} \Longrightarrow \begin{smallmatrix} abbb \\ babb \\ bbab \\ bbba \end{smallmatrix} \Longrightarrow \cdots.$$

The derivation proceeds deterministically using the RC-type rule. A language generated by γ is a set of squares whose positions in the main diagonal are covered by a and the remaining ones are covered by b.

Lemma 1 *There is a picture language which cannot be generated by any picture insertion systems.*

Proof Consider a picture language defined by $(a^{2n+1})^{n\ominus} \ominus (a^n b a^n) \ominus (a^{2n+1})^{n\ominus}$ for $n \geq 1$.

The claim can be proved by contradiction. □

Lemma 2 *There is a picture insertion system γ such that $L(\gamma)$ is not generated by a tiling system.*

Proof Consider a picture insertion system $\gamma = (T, \{(\lambda, \begin{smallmatrix} ab \\ ab \end{smallmatrix}, \lambda)\}, \{\lambda\})$ with $T = \{a, b\}$. From the definition of γ, a picture p in $L(\gamma)$ satisfies that the number of a in p is equivalent to that of b.

Suppose that there is a tiling system $\mathcal{T} = (T, \Gamma, \theta, \pi)$ such that $L(\gamma) = L(\mathcal{T})$, where Γ is a finite alphabet, θ is a finite set of tiles over Γ, and $\pi : \Gamma \to T$ is a projection. Then we can generate a contradiction. □

Lemma 3 *For any $i, j \geq 0$, $INS_{*,*}^{i,j}$ is incomparable with TS.*

Proof As an example, for the class of picture insertion systems, we consider $INS_{*,*}^{0,0}$.

From Lemma 2, we can prove that there is a picture language $L(\gamma)$ in $INS_{*,*}^{0,0}$ but not in TS.

Consider a tiling system $\mathcal{T} = (\{a, b\}, \{a, b\}, \theta, \pi)$, where $\theta = \{ \begin{smallmatrix} ab \\ ba \end{smallmatrix}, \begin{smallmatrix} ba \\ ab \end{smallmatrix} \}$, $\pi : \{a, b\} \to \{a, b\}$ is an identity projection such that $\pi(x) = x$ with $x \in \{a, b\}$. The followings are some examples of pictures in $L(\mathcal{T})$; $\begin{smallmatrix} ab \\ ba \end{smallmatrix}, \begin{smallmatrix} aba \\ bab \end{smallmatrix}, \begin{smallmatrix} ab \\ ba \\ ab \end{smallmatrix}, \begin{smallmatrix} aba \\ bab \\ aba \end{smallmatrix}, \ldots$.

Suppose that there is a picture insertion system γ such that $L(\gamma) = L(\mathcal{T})$, then we can generate a contradiction.

Similarly, for the case of $INS_{*,*}^{i,j}$ with $i, j \geq 0$, the claim can be proved. □

4 Concluding Remarks

In this paper, we introduced picture insertion systems which generate two-dimensional languages.

As considered in 1D 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, representation theorems are shown in 1D case [1]. Those representation might be possible in 2D case. Furthermore, to compare with cellular automaton is also our future work.

References

[1] K. Fujioka. Morphic characterizations of languages in Chomsky hierarchy with insertion and locality. *Information and Computation* **209**, pp.397–408, 2011.

[2] G. Păun, G. Rozenberg, and A. Salomaa. *DNA Computing: new computing paradigms.* Springer, 1998.

[3] S.C. Reghizzi and M. Pradella. Tile rewriting grammars and picture languages. *Theoretical Computer Science*, **340**, pp.257–272, 2005.

[4] G. Rozenberg and A. Salomaa, Eds. *Handbook of Formal Languages.* Springer, 1997.