Recognizability of Partial Array Languages

F. SWEETY[†], D.G. THOMAS[‡], V.R. DARE[‡] AND T. KALYANI[†]
[†] Department of Mathematics
St. Joseph's College of Engineering
Chennai - 600119, India.
[‡] Department of Mathematics
Madras Christian College
Chennai - 600059, India.
e-mail: *dgthomasmcc@yahoo.com*

Abstract

In this paper, we introduce an online tessellation partial automaton to recognize partial array languages. We also introduce two classes of partial array languages viz, Local Partial Array Languages (PAL-LOC), Recognizable Partial Array Languages (PAL-REC) and proved PAL-REC is exactly the family of partial array languages recognizable by online tessellation partial automaton.

Keywords. Partial array, recognizability, online tessellation partial automaton (OTPA).

1 Introduction

Aldo de Luca [1] introduced the combinatorial method for the analysis of finite words for the study of biological molecules and the partial words were introduced by Berstel and Boasson [2] in the context of gene (or protein) comparison. To develop an overall picture of how genes are regulated during hyphal development, the partial DNA array are used to study difference in gene expressions between wild type and the signaling mutants. In [8] we introduced a combinatorial method for the analysis of finite partial arrays. In this paper we introduce local partial array languages and recognizable partial array languages and we construct an Online tessellation Partial Automaton (OTPA). Using this OTPA, any living tissue can be read by considering each cell as an array. The main motivation for the introduction of partial array comes from molecular biology of nucleic acids. There are several extensions to two-dimensional cases of the well-known

JCMCC **69** (2009), pp. 237-249

notion of recognizability in terms of automata [5]. The first generalization of finite state automata to two dimensions is given by Blum and Hewitt [3], who introduced the notion of four-way automaton. An interesting model of two-dimensional tape acceptor is the two-dimensional online tessellation automaton introduced by Inoue and Nakamura [7]. This is a special type of a cellular automaton. Giammarresi and Restivo have provided a definition of recognizability in terms of local array languages [6]. In [9] we constructed a three dimensional online tessellation automaton 3-OTA to recognize 3D picture languages. This notion of recognizability is also established by different formalism namely 3D domino system and we proved that the family of 3D languages recognized by 3D domino system coincides with 3D recognizable languages [10]. In this paper we have extended the concept of recognizability of picture languages to partial array languages and proved that the family of recognizable partial array languages coincides with the family of languages recognizable by online tessellation partial automata.

2 Basic Notations and Definitions

In this section we deal with basic concepts on arrays, partial words and partial arrays.

Definition 2.1. *Let Σ be a finite alphabet. A rectangular array over Σ of size (m, n) is a rectangular arrangement of elements of Σ, having m rows and n columns, where $m, n \geq 0$.*

The set of all arrays over Σ (including an empty array Λ) is denoted by Σ^{**} and $\Sigma^{++} = \Sigma^{**} - \{\Lambda\}$.

Definition 2.2. *Let Σ be a finite alphabet and Σ^* be the collection of all words over Σ. A word of length n over Σ can be defined by a total function, $u : \{1, 2, \ldots, n\} \to \Sigma$ and is usually represented as $u = a_1 a_2 \ldots a_n$ with $a_i = u(i)$. The length of u is denoted by $|u|$. The word of length '0' is called the empty word and it is denoted by λ.*

Definition 2.3. *A partial word u of length n over Σ is a partial function $u : N \to \Sigma$, where n is the set of all natural numbers. For $1 \leq i \leq n$, if $u(i)$ is defined, then we say that i belongs to the domain of u (denoted by $i \in D(u)$), otherwise, we say that i belongs to the set of holes of u (denoted by $i \in H(u)$). A word over Σ is a partial word over Σ with an empty set of holes.*

Definition 2.4. *If u is a partial word of length n over Σ, then the companion of u (denoted by $u_\Diamond$) is the total function $u_\Diamond : N \to \Sigma \cup \{\Diamond\}$ defined* *by*

$$u_\Diamond(i) = \begin{cases} u(i) & \text{if } i \in D(u) \\ \Diamond & \text{otherwise, where } \Diamond \notin \Sigma \end{cases}$$

The symbol '$\Diamond$' is viewed as a 'do not know' symbol and not as a 'do not care' symbol as in pattern matching.

Definition 2.5. *A partial array A of size (m,n) over Σ is a partial function $A : Z^2 \to \Sigma$. For $1 \le i \le m$, $1 \le j \le n$, if $A(i,j)$ is defined then we say that (i,j) belongs to the domain of A (denoted by $(i,j) \in D(A)$), otherwise we say that (i,j) belongs to the set of holes of A (denoted by $(i,j) \in H(A)$). An array over Σ is a partial array over Σ with an empty set of holes.*

Definition 2.6. *If A is a partial array of size (m,n) over Σ then the companion of A (denoted by $A_\Diamond$) is the total function $A_\Diamond : Z^2 \to \Sigma \cup \{\Diamond\}$ defined by*

$$A_\Diamond(i,j) = \begin{cases} A(i,j) & \text{if } (i,j) \in D(A) \\ \Diamond & \text{otherwise, where } \Diamond \notin \Sigma \end{cases}$$

The symbol '$\Diamond$' is viewed as a 'do not know' symbol and not as a 'do not care' symbol as in pattern matching.

Example 2.1. *The partial array* $A_\Diamond = \begin{pmatrix} a & b & a \\ \Diamond & b & a \\ a & \Diamond & b \end{pmatrix}$ *is a companion of partial array A of size $(3,3)$ where*
$D(A) = \{(1,1),(1,3),(2,2),(2,3),(2,2),(2,3),(3,1),(3,2),(3,3)\}$ *and*
$H(A) = \{(1,2),(2,1)\}$.

Definition 2.7. *If A and B are two partial arrays of equal size then A is contained in B, denoted by $A \subset B$ if $D(A) \subseteq D(B)$ and $A(i,j) = B(i,j)$ for all $(i,j) \in D(A)$. The partial arrays A and B are compatible denoted by $A \uparrow B$ if there exists a partial array C such that $A \subset C$ and $B \subset C$.*

As an example $A_\Diamond = \begin{pmatrix} a & b & a \\ \Diamond & b & a \\ a & \Diamond & b \end{pmatrix}$ and $B_\Diamond = \begin{pmatrix} \Diamond & b & \Diamond \\ a & b & a \\ a & \Diamond & b \end{pmatrix}$ are the companions of two partial arrays A and B that are compatible.

The set of all partial arrays is denoted by Σ_p^{**}, where $\Sigma_p = \Sigma \cup \{\Diamond\}$. We denote the empty array with no symbols by Λ and $\Sigma_p^{++} = \Sigma_p^{**} - \{\Lambda\}$. The set of all partial arrays over Σ of sizes (k,r), $k \le m$, $r \le n$ is denoted by $\Sigma_p^{k \times r}$.

Given any finite partial array A, we denote by $B_{k,r}(A)$, the set of all subarrays of A of size (k,r) such that if A is of size (m,n), then $k \le m, r \le$

n. For any array $A \in \Sigma_p^{**}$, of size (m, n) we denote by $\hat{A}$, the array of size $(m+2, n+2)$ obtained by surrounding A by a special boundary symbol $\# \notin \Sigma$. We call a tile a square array or partial array of size $(2, 2)$.

Example 2.2. *Let $A \in \Sigma_p^{**}$, then $B_{2,2}(\hat{A})$ is the collection of all subarrays or partial subarrays of $\hat{A}$ of size $(2, 2)$. For example if*

$$A = \begin{pmatrix} a & b & a \\ \Diamond & b & a \\ a & \Diamond & b \end{pmatrix}, \textit{then,}$$

$$\hat{A} = \begin{matrix} \# & \# & \# & \# & \# \\ \# & a & b & a & \# \\ \# & \Diamond & b & a & \# \\ \# & a & \Diamond & b & \# \\ \# & \# & \# & \# & \# \end{matrix} \quad \textit{and}$$

$$B_{2,2}(\hat{A}) = \left\{ \begin{array}{|c|c|} \hline \# & \# \\ \hline \# & a \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & \# \\ \hline a & b \\ \hline \end{array}, \begin{array}{|c|c|} \hline a & b \\ \hline \Diamond & b \\ \hline \end{array}, \ldots \right\}$$

3 Online Tessellation Partial Automata (OTPA)

Inoue and Nakamura [7] have introduced the notion of online tessellation automata to accept finite array languages and it was extended to infinite array languages by V.R. Dare et al. [4]. In this section we introduce the notion of acceptance of finite partial array languages by online tessellation partial finite automata.

Definition 3.1. *A deterministic (nondeterministic) online tessellation partial automaton is $\mathcal{M} = (\Sigma_p, Q, Q_h, Q_0, F, \delta)$ where $\Sigma_p = \Sigma \cup \{\Diamond\}$ is an input alphabet, Q is a finite set of states, Q_h is a finite set of hole states, $Q_0 \subseteq Q$ is a set of initial states, $F \subseteq Q \cup Q_h$ is a set of final states and $\delta : (Q \cup Q_h) \times (Q \cup Q_h) \times \Sigma_p \to (Q \cup Q_h)(2^{Q \cup Q_h})$ is a transition function and it is defined as follows:*

(i) $\delta(q_1, q_2, a) = q_3 : q_1, q_2, q_3 \in Q,\ a \in \Sigma$

(ii) $\delta(q, q_h, a) = q_1 : q, q_1 \in Q,\ q_h \in Q_h,\ a \in \Sigma$

(iii) $\delta(q_h, q, a) = q_1 : q, q_1 \in Q,\ q_h \in Q_h,\ a \in \Sigma$

(iv) $\delta(q_{h_1}, q_{h_2}, a) = q : q \in Q,\ q_{h_1}, q_{h_2} \in Q_h,\ a \in \Sigma$

(v) $\delta(q_1, q_2, \Diamond) = q_h : q_1, q_2 \in Q,\ q_h \in Q_h$

(vi) $\delta(q, q_h, \Diamond) = q_{h_1} : q \in Q,\ q_h, q_{h_1} \in Q_h$

(vii) $\delta(q_h, q, \Diamond) = q_{h_1} : q \in Q,\ q_h, q_{h_1} \in Q_h$

(viii) $\delta(q_{h_1}, q_{h_2}, \Diamond) = q_{h_3} : q_{h_1}, q_{h_2}, q_{h_3} \in Q_h$

A computation by a two-dimensional OTPA on a finite partial array A of size $m \times n$ where

$$\hat{A}_\Diamond = \begin{matrix} \# & \# & \# & \cdots & \# & \# \\ \# & a_{m1} & a_{m2} & \cdots & a_{mn} & \# \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ \# & a_{21} & a_{22} & \cdots & a_{2n} & \# \\ \# & a_{11} & a_{12} & \cdots & a_{1n} & \# \\ \# & \# & \# & \cdots & \# & \# \end{matrix}$$

with $a_{ij} \in \Sigma_p$ and $\#$ is a special symbol not in Σ is done as follows.

At time $t = 0$, an initial state $q_0 \in Q_0$ is associated with all the positions of $\hat{A}$ holding $\#$. The state associated with each position (i, j) by the transition function δ, depends on the states already associated with the positions $(i, j-1)$, $(i-1, j)$ and the symbol a_{ij}. Let s_{ij} be the state associated with the position (i, j) if $a_{ij} \in \Sigma$ and s_{h_j} be the state associated with the position (i, j) if $a_{ij} = \Diamond$. At time $t = 1$, a state from $\delta(q_0, q_0, a_{11})$ is associated with the position $(1, 1)$ holding a_{11}. If $a_{11} \in \Sigma$ then, s_{11} is the state associated with the position $(1, 1)$ and if $a_{11} = \Diamond$, then s_{h_1} is the state associated with the position $(1, 1)$. At time $t = 2$, states are associated simultaneously with positions $(2, 1)$ and $(1, 2)$ respectively holding a_{21} and a_{12}.

Case i. If s_{11} is the state associated with the position $(1, 1)$ then the state associated with the position $(2, 1)$is an element of $\delta(q_0, s_{11}, a_{21})$ and to the position $(1, 2)$ is an element of $\delta(s_{11}, q_0, a_{12})$. If $a_{21} \in \Sigma$, then $\delta(q_0, s_{11}, a_{21}) = s_{21}$ and if $a_{21} = \Diamond$ then $\delta(q_0, s_{11}, \Diamond) = s_{h_2}$.

Similarly $\delta(s_{11}, q_0, a_{12}) = s_{12}$ if $a_{12} \in \Sigma$ if not $\delta(s_{11}, q_0, \Diamond) = s_{h_2}$.

Case ii. If s_{h_1} is the state associated with the position $(1, 1)$ then the state associated with the position $(2, 1)$ is an element of $\delta(q_0, s_{h_1}, a_{21})$ and to the position $(1, 2)$ is an element of $\delta(s_{h_1}, q_0, a_{12})$, and they are given as follows:

(i) $\delta(q_0, s_{h_1}, a_{21}) = s_{21}$ if $a_{21} \in \Sigma$

(ii) $\delta(q_0, s_{h_1}, \Diamond) = s_{h_2}$

(iii) $\delta(s_{h_1}, q_0, a_{12}) = s_{12}$ if $a_{12} \in \Sigma$

(iv) $\delta(s_{h_1}, q_0, \Diamond) = s_{h_2}$.

We then proceed to the next diagonal. The automaton stops its computation by reading the symbol a_{mn} and associating the state s_{mn}, if $a_{mn} = \Diamond$ then s_{h_n} is the state associated. A run (or a computation) of a finite partial array is an element of $(Q \cup Q_h)^{mn}$. A run for a finite partial array is a sequence of states

$$s_{11}s_{21}s_{12}s_{31}s_{22}s_{13}\ldots \text{ where } s_{ij} \in Q \cup Q_h$$

and it is denoted by $r(\mathcal{M})$. The language of finite partial arrays recognized by the non deterministic online tessellation automaton $\mathcal{M}$ is denoted by $L(\mathcal{M})$ and let $\mathcal{L}(OTPA)$ be the set of all partial array languages recognized by OTPA's.

Example 3.1. *Let*

$$\hat{A}_\Diamond = \begin{matrix} \# & \# & \# & \# & \# & \# \\ \# & a & b & \Diamond & b & \# \\ \# & a & b & a & \Diamond & \# \\ \# & \Diamond & b & a & b & \# \\ \# & a & \Diamond & b & a & \# \\ \# & \# & \# & \# & \# & \# \end{matrix}$$

An OTPA to recognize the above partial array A with boundary symbols is given by

$\mathcal{M} = (\Sigma_p, Q, Q_h, Q_0, F, \delta)$

$\Sigma = \{a, b\};$ $\quad Q = \{q_0, q_1, q_2\}$

$Q_0 = \{q_0\};$ $\quad F = \{q_2\}$

$Q_h = \{q_{h_1}, q_{h_2}, q_{h_3}, q_{h_4}\}$

$\delta(q_0, q_0, a) = q_1;$	$\delta(q_0, q_1, \Diamond) = q_{h_2}$
$\delta(q_1, q_0, \Diamond) = q_{h_1};$	$\delta(q_0, q_{h_2}, a) = q_1$
$\delta(q_{h_2}, q_{h_1}, b) = q_2;$	$\delta(q_{h_1}, q_0, b) = q_2$
$\delta(q_0, q_1, a) = q_1;$	$\delta(q_1, q_2, b) = q_2$
$\delta(q_2, q_2, a) = q_1;$	$\delta(q_2, q_0, a) = q_1$
$\delta(q_1, q_2, b) = q_2;$	$\delta(q_2, q_1, a) = q_1$
$\delta(q_1, q_1, b) = q_2;$	$\delta(q_2, q_1, \Diamond) = q_{h_4}$
$\delta(q_1, q_2, \Diamond) = q_{h_3};$	$\delta(q_{h_4}, q_{h_3}, b) = q_2$

4 Local and Recognizable Partial Array Languages

In this section we introduce local partial array languages, recognizable partial array languages and establish a relation between $\mathcal{L}(OTPA)$ and recognizable partial array languages.

Definition 4.1. *Let Γ_p be a finite alphabet where $\Gamma_p = \Gamma \cup \{\Diamond\}$. A two-dimensional partial array language $L \subseteq \Gamma_p^{**}$ is local if there exist a finite set θ of tiles over the alphabet $\Gamma_p \cup \{\#\}$ such that $L = \{A \in \Gamma_p^{**} / B_{2,2}(\hat{A}) \subseteq \theta\}$.*

Therefore θ represents the set of blocks or hollow blocks for pictures belonging to the local language L. The language L is local if, given such a set θ, we can exactly retrieve the language L. We call the set θ a representation by tiles for the local language L and write $L = L(\theta)$.

Example 4.1. *Let $\Gamma_p = \{a, b\} \cup \{\Diamond\}$ be an alphabet and let θ be the following set of tiles over Γ_p.*

$$\theta = \left\{ \begin{array}{|c|c|}\hline \# & \# \\ \hline \# & b \\ \hline \end{array}, \begin{array}{|c|c|}\hline \# & \# \\ \hline b & b \\ \hline \end{array}, \begin{array}{|c|c|}\hline \# & \# \\ \hline b & \# \\ \hline \end{array}, \begin{array}{|c|c|}\hline \# & b \\ \hline \# & a \\ \hline \end{array}, \begin{array}{|c|c|}\hline b & b \\ \hline a & \Diamond \\ \hline \end{array}, \begin{array}{|c|c|}\hline b & b \\ \hline \Diamond & b \\ \hline \end{array}, \right.$$
$$\begin{array}{|c|c|}\hline b & \# \\ \hline b & \# \\ \hline \end{array}, \begin{array}{|c|c|}\hline \# & a \\ \hline \# & a \\ \hline \end{array}, \begin{array}{|c|c|}\hline a & \Diamond \\ \hline a & a \\ \hline \end{array}, \begin{array}{|c|c|}\hline \Diamond & b \\ \hline a & b \\ \hline \end{array}, \begin{array}{|c|c|}\hline a & a \\ \hline \# & \# \\ \hline \end{array}, \begin{array}{|c|c|}\hline a & b \\ \hline \# & \# \\ \hline \end{array},$$
$$\begin{array}{|c|c|}\hline \# & a \\ \hline \# & \# \\ \hline \end{array}, \begin{array}{|c|c|}\hline b & \# \\ \hline \# & \# \\ \hline \end{array}, \begin{array}{|c|c|}\hline b & b \\ \hline \Diamond & \Diamond \\ \hline \end{array}, \begin{array}{|c|c|}\hline a & \Diamond \\ \hline a & \Diamond \\ \hline \end{array}, \begin{array}{|c|c|}\hline \Diamond & \Diamond \\ \hline \Diamond & \Diamond \\ \hline \end{array}, \begin{array}{|c|c|}\hline \Diamond & b \\ \hline \Diamond & b \\ \hline \end{array},$$
$$\left. \begin{array}{|c|c|}\hline b & \# \\ \hline b & \# \\ \hline \end{array}, \begin{array}{|c|c|}\hline \Diamond & \Diamond \\ \hline a & a \\ \hline \end{array}, \begin{array}{|c|c|}\hline \# & a \\ \hline \# & a \\ \hline \end{array} \right\}$$

Then $L(\theta)$ is a partial array language over Γ_p with equal sides of length, the symbols along the first row and last column are b's, the symbols along the last row and the first column except the first and last elements of the principal diagonals are a's. The remaining elements of the array are holes.

The first two members of this language are given below

$$\begin{array}{ccc} b & b & b \\ a & \Diamond & b \\ a & a & b \end{array}, \quad \begin{array}{cccc} b & b & b & b \\ a & \Diamond & \Diamond & b \\ a & \Diamond & \Diamond & b \\ a & a & a & b \end{array}, \quad \ldots$$

Definition 4.2. *Let Σ be a finite alphabet. A partial array language $L \subseteq \Sigma_p^{**}$ is called recognizable if there exists a local partial array language L' over an alphabet Γ_p and a mapping $\pi : \Gamma_p \to \Sigma_p$ such that $L = \pi(L')$.*

Example 4.2. *The set of all partial array languages over one letter alphabet 'a' with all sides of equal length and the symbols along the first row, first column, the last row and last column are holes is not a local partial array language, but it is a recognizable partial language. This language is obtained from Example 4.1 by taking a mapping $\pi : \Gamma_p \to \Sigma_p$ where $\Gamma = \{a, b\}$, $\Sigma = \{a\}$ such that $\pi(b) = \pi(a) = \Diamond$ and $\pi(\Diamond) = a$.*

The family of all recognizable partial array languages is denoted by PAL-REC.

Definition 4.3. *A partial array tiling system (PATS) is a 4-tuple* $(\Sigma_p, \Gamma_p, \pi, \theta)$ *where* Σ *and* Γ *are two finite sets of symbols,* $\pi : \Gamma_p \to \Sigma_p$ *is a projection and* θ *is a set of partial blocks or hollow blocks over the alphabet* $\Gamma_p \cup \{\#\}$*. The partial array language* $L \subseteq \Sigma_p^{**}$ *is tiling recognizable if there exists a tiling system* $T = (\Sigma_p, \Gamma_p, \pi, \theta)$ *such that* $L = \pi(L(\theta))$*. It is denoted by* $L(T)$*. The family of partial language recognizable by a partial array tiling system is denoted by* $\mathcal{L}(PATS)$*.*

It is easy to see that PAL-REC is exactly the family of partial array languages recognizable by partial array tiling system ($\mathcal{L}(PATS)$)*.*

Theorem 4.1. *The class of all partial array languages recognized by online tessellation partial automaton coincides with PAL-REC.*

The proof of this theorem follows from the following lemmas.

Lemma 4.1. *If a partial array language is recognizable by online tessellation partial automaton then it is recognizable by partial array tiling system.*

Proof. Let $L \in \Sigma_p^{**}$ be a language recognized by an online tessellation automaton $\mathcal{M} = (\Sigma, Q, q_0, q_h, \delta, F)$, then there exist a partial array tiling system T, that recognizes L.

Let $T = (\Sigma_p, \Gamma_p, Q, \pi)$ be a tiling system such that, $\Gamma_p = (\Sigma_p \cup \{\#\}) \times (Q \cup Q_h)$ and
$\pi : (\Sigma_p \cup \{\#\}) \times (Q \cup Q_h) \to \Sigma_p$ is such that
$\pi(a, q) = a$ where $q \in Q \cup Q_h$, $a \in \Sigma$
$\pi(\Diamond, q) = \Diamond$ where $q \in Q \cup Q_h$

$$\theta_1 = \left\{ \begin{array}{|c|c|} \hline (\#, q_0) & (a, r) \\ \hline (\#, q_0) & (\#, q_0) \\ \hline \end{array} \Bigg/ \begin{array}{l} a \in \Sigma \\ r \in \delta(q_0, q_0, a) \end{array} \right\}$$

$$\theta_2 = \left\{ \begin{array}{|c|c|} \hline (\#, q_0) & (\Diamond, q_h) \\ \hline (\#, q_0) & (\#, q_0) \\ \hline \end{array} \Bigg/ q_h \in \delta(q_0, q_0, \Diamond) \right\}$$

$$\theta_3 = \left\{ \begin{array}{|c|c|} \hline (\#, q_0) & (b, s) \\ \hline (\#, q_0) & (a, r) \\ \hline \end{array} \Bigg/ \begin{array}{l} a, b \in \Sigma \\ s \in \delta(q_0, r, b) \end{array} \right\}$$

$$\theta_4 = \left\{ \begin{array}{|c|c|} \hline (\#, q_0) & (b, s) \\ \hline (\#, q_0) & (\Diamond, q_h) \\ \hline \end{array} \Bigg/ \begin{array}{l} b \in \Sigma \\ s \in \delta(q_0, q_h, b) \end{array} \right\}$$

$$\theta_5 = \left\{ \begin{array}{|c|c|} \hline (\#, q_0) & (\Diamond, q_h) \\ \hline (\#, q_0) & (a, r) \\ \hline \end{array} \Bigg/ \begin{array}{l} a \in \Sigma \\ q_h \in \delta(q_0, r, \Diamond) \end{array} \right\}$$

$$\theta_6 = \left\{ \begin{array}{|c|c|} \hline (\#, q_0) & (\Diamond, q_h) \\ \hline (\#, q_0) & (\Diamond, q_h) \\ \hline \end{array} \Bigg/ q_h \in \delta(q_0, q_h, \Diamond) \right\}$$

$$\theta_7 = \left\{ \begin{array}{|c|c|} \hline (\#, q_0) & (\#, q_0) \\ \hline (\#, q_0) & (c, t) \\ \hline \end{array} \Bigg/ \begin{array}{l} c \in \Sigma \\ q_0 \in \delta(q_0, t, \#) \end{array} \right\}$$

$$\theta_8 = \left\{ \begin{array}{|c|c|}\hline (\#,q_0) & (\#,q_0) \\ \hline (\#,q_0) & (\Diamond,q_h) \\ \hline \end{array} \Big/ \; q_0 \in \delta(q_0,q_h,\#) \right\}$$

$$\theta_9 = \left\{ \begin{array}{|c|c|}\hline (b,s) & (a,r) \\ \hline (\#,q_0) & (\#,q_0) \\ \hline \end{array} \Big/ \; \begin{array}{l} a,b \in \Sigma \\ r \in \delta(s,q_0,a) \end{array} \right\}$$

$$\theta_{10} = \left\{ \begin{array}{|c|c|}\hline (b,s) & (\Diamond,q_h) \\ \hline (\#,q_0) & (\#,q_0) \\ \hline \end{array} \Big/ \; \begin{array}{l} b \in \Sigma \\ q_h \in \delta(s,q_0,\Diamond) \end{array} \right\}$$

$$\theta_{11} = \left\{ \begin{array}{|c|c|}\hline (\Diamond,q_h) & (a,r) \\ \hline (\#,q_0) & (\#,q_0) \\ \hline \end{array} \Big/ \; \begin{array}{l} a \in \Sigma \\ r \in \delta(q_h,q_0,a) \end{array} \right\}$$

$$\theta_{12} = \left\{ \begin{array}{|c|c|}\hline (\Diamond,q_h) & (\Diamond,q_h) \\ \hline (\#,q_0) & (\#,q_0) \\ \hline \end{array} \Big/ \; q_h \in \delta(q_h,q_0,\Diamond) \right\}$$

$$\theta_{13} = \left\{ \begin{array}{|c|c|}\hline (c,t) & (d,u) \\ \hline (a,r) & (b,s) \\ \hline \end{array} \Big/ \; \begin{array}{l} a,b,c,d \in \Sigma \\ u \in \delta(t,s,d) \end{array} \right\}$$

$$\theta_{14} = \left\{ \begin{array}{|c|c|}\hline (c,t) & (d,u) \\ \hline (a,r) & (\Diamond,q_h) \\ \hline \end{array} \Big/ \; \begin{array}{l} a,c,d \in \Sigma \\ u \in \delta(t,q_h,d) \end{array} \right\}$$

$$\theta_{15} = \left\{ \begin{array}{|c|c|}\hline (c,t) & (d,u) \\ \hline (\Diamond,q_h) & (b,s) \\ \hline \end{array} \Big/ \; \begin{array}{l} b,c,d \in \Sigma \\ u \in \delta(t,s,d) \end{array} \right\}$$

$$\theta_{16} = \left\{ \begin{array}{|c|c|}\hline (\Diamond,q_h) & (d,u) \\ \hline (a,r) & (b,s) \\ \hline \end{array} \Big/ \; \begin{array}{l} a,b,d \in \Sigma \\ u \in \delta(q_h,s,d) \end{array} \right\}$$

$$\theta_{17} = \left\{ \begin{array}{|c|c|}\hline (c,t) & (d,u) \\ \hline (\Diamond,q_h) & (\Diamond,q_h) \\ \hline \end{array} \Big/ \; \begin{array}{l} c,d \in \Sigma \\ u \in \delta(t,q_h,d) \end{array} \right\}$$

$$\theta_{18} = \left\{ \begin{array}{|c|c|}\hline (\Diamond,q_h) & (d,u) \\ \hline (\Diamond,q_h) & (b,s) \\ \hline \end{array} \Big/ \; \begin{array}{l} b,d \in \Sigma \\ u \in \delta(t,q_h,d) \end{array} \right\}$$

$$\theta_{19} = \left\{ \begin{array}{|c|c|}\hline (\Diamond,q_h) & (d,u) \\ \hline (a,r) & (\Diamond,q_h) \\ \hline \end{array} \Big/ \; \begin{array}{l} a,d \in \Sigma \\ u \in \delta(q_h,q_h,d) \end{array} \right\}$$

$$\theta_{20} = \left\{ \begin{array}{|c|c|}\hline (\Diamond,q_h) & (d,u) \\ \hline (\Diamond,q_h) & (\Diamond,q_h) \\ \hline \end{array} \Big/ \; \begin{array}{l} d \in \Sigma \\ u \in \delta(q_h,q_h,d) \end{array} \right\}$$

$\theta = \bigcup_{i=1}^{20} \theta_i$

Let L' be the local partial array language corresponding to the set θ defined above, then it is easy to verify that $\pi(L') = L$. Hence L is PAL-REC. □

Lemma 4.2. *If a language is recognizable by partial array tiling system then it is recognizable by an online tessellation partial automaton.*

Proof. Let $L \subseteq \Sigma_p^{**}$ be a partial array language recognized by partial array tiling system $(\Sigma_p, \Gamma_p, \theta, \pi)$ and L' be a local partial array language represented by the set θ of blocks (or) hollow blocks then $\pi(L') = L$. It suffices to show that there exist an OTPA recognizing $L' \subseteq \Gamma_p^{**}$. □

Lemma 4.3. *If L is a local partial array language then it is recognizable by an OTPA.*

Proof. Let $L \subseteq \Sigma_p^{**}$ be a local partial array language. Then $L = L(\Delta)$ where Δ is a set of blocks or hollow blocks over $\Sigma_p \cup \{\#\}$.

We construct an OTPA as follows:

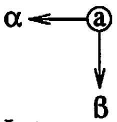
$\mathcal{M} = \{\Sigma_p, Q, Q_h, I, \delta, F\}$, where $Q \cup Q_h = \Delta$

$$I = \left\{ \begin{array}{|c|c|}\hline \# & a \\ \hline \# & \# \\ \hline \end{array}, \begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & \# \\ \hline \end{array} \Big/ a \in \Sigma, \right\}$$

$$F = \left\{ \begin{array}{|c|c|}\hline \# & \# \\ \hline a & \# \\ \hline \end{array}, \begin{array}{|c|c|}\hline \# & \# \\ \hline \Diamond & \# \\ \hline \end{array} \Big/ a \in \Sigma \right\}$$

The transition $\delta : Q \cup Q_h \times Q \cup Q_h \times \Sigma_p \to 2^{Q \cup Q_h}$ is defined in a way that the run of $\mathcal{M}$ over a partial array A simulates a tiling of A by elements of $Q \cup Q_h = \Delta$ is given as follows:

Given a bordered partial array $\hat{A}$ with $A \in L(\Delta)$ for each symbol 'a' in the $(i,j)^{th}$ position of A, where $a \in \Sigma_p$, we first find two symbols α, β of $\Sigma_p \cup \{\#\}$ in $\hat{A}$, associated with a, using the following diagram



Let us consider the following cases.

Case i(a) If $a \in \Sigma$ and $\alpha = \beta = \#$, then

$$(1)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & a \\ \hline \end{array}, \begin{array}{|c|c|}\hline \# & a \\ \hline \# & \# \\ \hline \end{array}, a \right) = \begin{array}{|c|c|}\hline a & b \\ \hline \# & \# \\ \hline \end{array}$$

$$(2)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & a \\ \hline \end{array}, \begin{array}{|c|c|}\hline \# & a \\ \hline \# & \# \\ \hline \end{array}, a \right) = \begin{array}{|c|c|}\hline a & \Diamond \\ \hline \# & \# \\ \hline \end{array}$$

$$(3)\ \delta\left(\begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & a \\ \hline \end{array}, \begin{array}{|c|c|}\hline \# & a \\ \hline \# & \# \\ \hline \end{array}, a \right) = \begin{array}{|c|c|}\hline a & b \\ \hline \# & \# \\ \hline \end{array}$$

$$(4)\ \delta\left(\begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & a \\ \hline \end{array}, \begin{array}{|c|c|}\hline \# & a \\ \hline \# & \# \\ \hline \end{array}, a \right) = \begin{array}{|c|c|}\hline a & \Diamond \\ \hline \# & \# \\ \hline \end{array}$$

Case i(b) If $a = \Diamond$, $b \in \Sigma$ and $\alpha = \beta = \#$, then

$$(1)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & \Diamond \\ \hline \end{array}, \begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & \# \\ \hline \end{array}, \Diamond \right) = \begin{array}{|c|c|}\hline \Diamond & b \\ \hline \# & \# \\ \hline \end{array}$$

$$(2)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & \Diamond \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & \# \\ \hline\end{array}, \Diamond\right) = \begin{array}{|c|c|}\hline \Diamond & \Diamond \\ \hline \# & \# \\ \hline\end{array}$$

$$(3)\ \delta\left(\begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & \Diamond \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & \# \\ \hline\end{array}, \Diamond\right) = \begin{array}{|c|c|}\hline \Diamond & b \\ \hline \# & \# \\ \hline\end{array}$$

$$(4)\ \delta\left(\begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & \Diamond \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & \# \\ \hline\end{array}, \Diamond\right) = \begin{array}{|c|c|}\hline \Diamond & \Diamond \\ \hline \# & \# \\ \hline\end{array}$$

Case ii(a) If $a \in \Sigma$, $\alpha = \#$ and $\beta \neq \#$, then

$$(1)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & a \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & a \\ \hline \# & b \\ \hline\end{array}, a\right) = \begin{array}{|c|c|}\hline a & b \\ \hline c & d \\ \hline\end{array}$$

$$(2)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & a \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & a \\ \hline \# & b \\ \hline\end{array}, a\right) = \begin{array}{|c|c|}\hline a & \Diamond \\ \hline c & d \\ \hline\end{array}$$

$$(3)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & a \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & a \\ \hline \# & b \\ \hline\end{array}, a\right) = \begin{array}{|c|c|}\hline a & \Diamond \\ \hline c & \Diamond \\ \hline\end{array}$$

$$(4)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & a \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & a \\ \hline \# & b \\ \hline\end{array}, a\right) = \begin{array}{|c|c|}\hline a & \Diamond \\ \hline \Diamond & \Diamond \\ \hline\end{array}$$

$$(5)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & a \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & a \\ \hline \# & b \\ \hline\end{array}, a\right) = \begin{array}{|c|c|}\hline a & b \\ \hline \Diamond & d \\ \hline\end{array}$$

$$(6)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & a \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & a \\ \hline \# & b \\ \hline\end{array}, a\right) = \begin{array}{|c|c|}\hline a & b \\ \hline c & \Diamond \\ \hline\end{array}$$

$$(7)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & a \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & a \\ \hline \# & b \\ \hline\end{array}, a\right) = \begin{array}{|c|c|}\hline a & b \\ \hline \Diamond & \Diamond \\ \hline\end{array}$$

$$(8)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & a \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & a \\ \hline \# & \Diamond \\ \hline\end{array}, a\right) = \begin{array}{|c|c|}\hline a & b \\ \hline \Diamond & c \\ \hline\end{array}$$

$$(9)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & a \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & a \\ \hline \# & \Diamond \\ \hline\end{array}, a\right) = \begin{array}{|c|c|}\hline a & \Diamond \\ \hline \Diamond & c \\ \hline\end{array}$$

$$(10)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & a \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & a \\ \hline \# & \Diamond \\ \hline\end{array}, a\right) = \begin{array}{|c|c|}\hline a & b \\ \hline \Diamond & \Diamond \\ \hline\end{array}$$

$$(11)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & a \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & a \\ \hline \# & \Diamond \\ \hline\end{array}, a\right) = \begin{array}{|c|c|}\hline a & \Diamond \\ \hline \Diamond & \Diamond \\ \hline\end{array}$$

Case ii(b) If $a = \Diamond$, $b, c, d \in \Sigma$, $\alpha = \#$, $\beta \neq \#$, then

$$(1)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & \Diamond \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & b \\ \hline\end{array}, \Diamond\right) = \begin{array}{|c|c|}\hline \Diamond & b \\ \hline c & d \\ \hline\end{array}$$

$$(2)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & \Diamond \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & b \\ \hline\end{array}, \Diamond\right) = \begin{array}{|c|c|}\hline \Diamond & \Diamond \\ \hline c & d \\ \hline\end{array}$$

$$(3)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & \Diamond \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & b \\ \hline\end{array}, \Diamond\right) = \begin{array}{|c|c|}\hline \Diamond & \Diamond \\ \hline c & \Diamond \\ \hline\end{array}$$

$$(4)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & \Diamond \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & b \\ \hline\end{array}, \Diamond\right) = \begin{array}{|c|c|}\hline \Diamond & \Diamond \\ \hline \Diamond & \Diamond \\ \hline\end{array}$$

$$(5)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & \Diamond \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & b \\ \hline\end{array}, \Diamond\right) = \begin{array}{|c|c|}\hline \Diamond & b \\ \hline \Diamond & d \\ \hline\end{array}$$

$$(6)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & \Diamond \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & b \\ \hline\end{array}, \Diamond\right) = \begin{array}{|c|c|}\hline \Diamond & b \\ \hline c & \Diamond \\ \hline\end{array}$$

$$(7)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & \Diamond \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & b \\ \hline\end{array}, \Diamond\right) = \begin{array}{|c|c|}\hline \Diamond & b \\ \hline \Diamond & \Diamond \\ \hline\end{array}$$

$$(8)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & \Diamond \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & \Diamond \\ \hline\end{array}, \Diamond\right) = \begin{array}{|c|c|}\hline \Diamond & b \\ \hline \Diamond & d \\ \hline\end{array}$$

$$(9)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & \Diamond \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & \Diamond \\ \hline\end{array}, \Diamond\right) = \begin{array}{|c|c|}\hline \Diamond & \Diamond \\ \hline \Diamond & d \\ \hline\end{array}$$

$$(10)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & \Diamond \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & \Diamond \\ \hline\end{array}, \Diamond\right) = \begin{array}{|c|c|}\hline \Diamond & b \\ \hline \Diamond & \Diamond \\ \hline\end{array}$$

$$(11)\ \delta\left(\begin{array}{|c|c|}\hline \# & b \\ \hline \# & \Diamond \\ \hline\end{array}, \begin{array}{|c|c|}\hline \# & \Diamond \\ \hline \# & \Diamond \\ \hline\end{array}, \Diamond\right) = \begin{array}{|c|c|}\hline \Diamond & \Diamond \\ \hline \Diamond & \Diamond \\ \hline\end{array}$$

Similarly we can do for the other combinations such as (i) $\alpha \neq \#$, but $\beta = \#, a \in \Sigma_p$ (ii) $\alpha \neq \#, \beta \neq \#, a \in \Sigma_p$. It can be easily verified that $L = L(\mathcal{M})$. □

Conclusion

In this paper we have introduced an online tessellation partial automaton and proved that the family of partial array languages is recognizable by this automaton.

References

[1] Aldo de Luca, On the combinatorics of finite words, *TCS* **218**(1999), 13-39.

[2] J. Berstel and L. Boasson, Partial words and a theorm of Fine and Wilf, *Theoretical Computer Science,* **218**(1999), 135–141.

[3] M. Blum and C. Hewitt, Automata on a 2-dimensional tape, *IEEE Symposium on Switching and Automata Theory,* (1967), 155-160, 1967.

[4] V.R. Dare, K.G. Subramanian, D.G. Thomas and R. Siromoney, Infinite arrays and Recognizability, *International Journal of Pattern Recognizition and Artificial Intelligence,* **14**(4)(2000), 525-536.

[5] D. Giammarresi and A. Restivo, Two dimensional finite state recognizability, *Fundamenta Informatica,* **25**(314)(1996), 399-422.

[6] D. Giammarresi and A. Restivo, *Two dimensional languages*, Handbook of Formal Languages, Vol. 3, Springer Verlag, Berlin, (1997), 215-265.

[7] K. Inoue and A. Nakamura, Some properties of two-dimensional online tessellation acceptors, *Information Sciences,* **13**(1977), 95-121.

[8] F. Sweety, D.G. Thomas and V.R. Dare, Subarray complexity of partial arrays, *Proc. of Intelligent Optimization Modeling*, Eds. P. Balasubramanian and R. Uthayakumar, Allied Publishers Pvt. Ltd., Chennai, (2006), 85-94.

[9] F. Sweety, D.G. Thomas, V.R. Dare and T. Kalyani, Automata for 3D Rectangular Picture Languages and Learning, *Proc. of the National Conference on Recent Advances in Computing (NCRAC 06)*, (2006), 83-92.

[10] F. Sweety, D.G. Thomas and T. Kalyani, Recognizability of 3D Rectangular Picture Languages by 3D Domino System, *Proc. of the Indian Conference on Intelligent System - ICIS 07,* (2007), 265-271.