Impact of Shuffle on Trajectories on Certain Classes of Partial array Languages

H. Geetha[1] K. Suresh[2] P. Venkatesh Babu[3] and T. Kalyani[4]
[1,2,4]Department of Mathematics,
[1,2] St. Joseph's College of Engineering,
[4] St. Joseph's Institute of Technology
Chennai-119, Tamilnadu, India.
[3] Department of Architecture,
SRM SEAD, SRMIST,
Chennai-89, Tamil Nadu, India.
E-mail: geethahvenki@hotmail.com, dhivasuresh@gmail.com,sp.v.babu@gmail.com

Abstract. In the recent years many tools are being used for developing picture languages. The vital and most interesting among such tools is the shuffle on trajectories. The above tool can be applied on various disciplines. In this paper,a Finite State Matrix Partial array Automaton is defined and someresults over Finite State Matrix Partial array Automaton and Online Tessellation Partial Automaton are studied.

Key Words: Finite State Matrix Partial Array automaton, online Tessellation Partial Automaton

1 Introduction

Berstel and Boasson [1] introduced the concept of partial words in their study on biological molecules. This concept of partial words was converted to arrays and thus the partial arrays were introduced and their combinatorial studies were also done. The recognizability of partial arrays using online tessellation automaton was studied in depth in [7]. The introduction of shuffle on trajectories and syntactic constraints were studied by mateescu et.al [5]. The concept of trajectories was applied to contextual array grammars in [4]. The shuffle on trajectories over finite array languages was studied [2]. Shuffle on array languages generated by array grammars was discussed in [6]. Inspired by the above studies, the tool shuffle on trajectories over partial array languages and its effects on certain classes of partial arrays were introduced [3].

In this paper we define Finite State Matrix Partial Array automaton and applied shuffle on trajectories to arrive two interesting theorems.

2 Preliminaries

In this section we give the definition of partial array and online Tessellation partial automaton.

Definition. 1

A deterministic online tessellation partial automaton (OTPA) is $\mathcal{B} = (\Sigma \cup \{\lozenge\}, Q, Q_h, q_0, F, \delta)$ where Q is a set of states, Q_h is the states associated with the holes $\lozenge$. $q_0 \in Q$ is the begining state. $F \subseteq (Q \cup Q_h)$ is a set of finishing states, the mapping δ function defined as $\delta : Q_1 \times Q_1 \times \Sigma \cup \{\lozenge\} \to Q_1$. Where $Q_1 = Q \cup Q_h$. For computation of a two-dimensional OTPA we refer [7]. The language of finite partial arrays recognized by OTPA, $\mathcal{B}$ is denoted by $L(\mathcal{B})$ and $\mathcal{L}(OTPA)$ is the set of partial array languages recognized by OTPA.

3 Main Results

In this section we define finite state matrix partial array automaton and a theorem on regular partial matrix array languages and on recognizable partial array Languages.

Definition. 2

A finite state matrix partial array automaton (FSMPA)is constructedas $\mathcal{B} = (Q_h, \Sigma \cup \{\lozenge\}, \Gamma, \delta, \delta', S_h, F_h, F_h', \$)$ where $Q_h = \bar{Q}_h \cup Q_{1h} \cup \ldots \cup Q_{kh}$, $Q_{ih} \cap Q_{jh} = \phi_1$ for $i \neq j$, that includes the states associated to $\lozenge$. Σ and Γ are the finite set of inputs and stack symbols respectively, with $\# |\Gamma| = k$. Each Q_{ih} has an initial state either $\{q_i\}$ or

ICASISET 2020, May 16-17, Chennai, India

DOI 10.4108/eai.16-5-2020.2304023

$\{q_{hi}\}$ and a final state $\{f_i\}$ or $\{f_{hi}\}$ $i=1,\cdots,k$. Define $F_h^{'}=\bigcup_{i=1}^{k}\left(\{f_i\}\cup\{f_{hi}\}\right)$ is the set of transition states. $S_h=\bigcup_{i=1}^{k}\left(\{q_i\}\cup\{q_{hi}\}\right)$ the set of start states. $\bar{Q}_h$ has an initial state q_0 and $F_h\subseteq\bar{Q}_h$ is the set of end states. $\$\not\subseteq\Sigma$ is to denote the end . δ is defined as $\delta: Q_{ih}\times\Sigma\cup\{\Diamond\}\rightarrow Q_{ih}\times\{\varepsilon\}$, $\{f_i\}\cup\{f_{hi}\}\times\$$ into $(S_h\cup\{q_0\})\times s_i$, here s_i is the stack withrespect to Q_{ih}. $\delta^{'}$ is defined from $\bar{Q}_h\times\Gamma$ into the finite subsets of $\bar{Q}_h$. Initially , the input matrix is placed at the end marker.

$$\begin{matrix} \$ & \ldots & \$ \\ b_{m1} & \ldots & b_{mn} \\ \ldots & \ldots & \ldots \\ b_{i1} & \ldots & b_{in} \end{matrix}$$

where $b_{ij}\in\Sigma\cup\{\Diamond\}, 1\le i\le m$ and $1\le j\le n$.

To begin with, the FSMPA starts recognizing from bottom to top as per δ transition, after reaching the first $\$$ it writes a symbol from Γ and then reaches the next column. This process is repeated untill the matrix of size $m\times n$ is read. At this stage the storage will have n symbols. Now, using $\delta^{'}$ from left to right and attains final state, if not the matrix is rejected .

The configuration $(p,(i,j),x,t)$ where p the present state in Q_h, (i,j) is the location of the input, x is a partial word, t is the number of positions from left.

If $(q,(m+1,n),x,r)$ is an arrangement and $\delta^{'}(q,z)$ has $q^{'}$ implies

$(q,(m+1,n),x,r)\underset{\delta^{'}}{\square}(q^{'},(m+1,n),z,r+1)$ where q belongs to $F_h^{'}$ and $q^{'}$ is in S_h.

Definition. 3

The set of all languages accepted by FSMPA is denoted as

$L(\mathsf{B})=\{[b_{ij}], i=1,2,\cdots m, j=1,2,\cdots n, m,n\ge 1 / b_{ij}\in\Sigma\cup\{\Diamond\},$

$(p,(1,1),\varepsilon,1)\underset{\delta}{\square}^{*}(q_0,(m+1,n),x,1)\underset{\delta^{'}}{\square}^{*}(p^{'},(m+1,n),x,n)$

with p in S_h, $p^{'}$ in F_h and x in Γ^{+} }

Note: The languages accepted by finite state matrix partial array automaton (FSMPA) are all regular partial matrix array languages.

Theorem. 1

L_1 and L_2 are the regular partial matrix array languages and if $T\subseteq\{r,u\}^{*}\cup\{l,d\}^{*}$, a regular set of trajectories, then $L_1 Ш_T L_2$ is also a regular partial matrix array language.

Proof.

Let $\mathsf{A}_1=(Q_{1h},\Sigma\cup\{\Diamond\},\Gamma_1,\delta_1,\delta_1^{'},S_{1h},F_{1h},F_{1h}^{'},\$)$ and

$\mathsf{A}_2=(Q_{2h},\Sigma\cup\{\Diamond\},\Gamma_2,\delta_2,\delta_2^{'},S_{2h},F_{2h},F_{2h}^{'},\$)$ are the two FSMPA and thus $L(\mathsf{A}_1)=L_1$ and $L(\mathsf{A}_2)=L_2$. Q_{1h} and Q_{2h} are defined respectively as

$Q_{1h}=\bar{Q}_{1h}\cup Q_{11h}\cup Q_{12h},\cdots,\cup Q_{1k_h}$, with $Q_{1i}\ne Q_{1j}$ and

$Q_{2h}=\bar{Q}_{2h}\cup Q_{21h}\cup Q_{22h},\cdots,\cup Q_{2kh}$ with $Q_{2i}\ne Q_{2j}$ for $i\ne j$,

Γ_1 and Γ_2 are the stack symbols and they correspond to one and only one Q_{1ih} and Q_{2ih} respectively. $|\Gamma_1|$ and $|\Gamma_2|$ each equal to k. Each Q_{1ih} and Q_{2ih} has astart state $\{q_{1i}\} or \{q_{h1i}\}$ and q_{2i} or q_{h2i}. Each with final states f_{1i} or f_{h1i} and f_{2i} or f_{h2i} where $i=1,2,\cdots,k$. $S_{1h}=\bigcup_{i=1}^{k}\{q_{1i}\cup q_{h1i}\}$ and $S_{2h}=\bigcup_{i=1}^{k}\{q_{2i}\cup q_{h2i}\}$ are the set of begining states. $F_{1h}^{'}=\bigcup_{i=1}^{k}\{f_{1i}\cup f_{h1i}\}$ and $F_{2h}^{'}=\bigcup_{i=1}^{k}\{f_{2i}\cup f_{h2i}\}$. $\bar{Q}_{1h}$ and $\bar{Q}_{2h}$ have starting states $\{q_{10}\cup q_{h10}\}$ and $\{q_{20}\cup q_{h20}\}$ respectively. $F_{1h}\subseteq\bar{Q}_{1h}, F_{2h}\subseteq\bar{Q}_{2h}$ are the end states. δ_1 is from $Q_{1ih}\times\Sigma\cup\{\Diamond\}$ into the finite subsets of $Q_{1ih}\times\{\grave{o}\}$, $i=1,2,\cdots,k$ and from $f_{1i}\times\$$ into $\left(S_{1h}\cup\{q_{10}\cup q_{h10}\}\right)\times s_{1i}$ where s_{1i} is the stack symbol corresponding to Q_{1i}. δ_1' is the mapping from $\bar{Q}_{1h} \times \Gamma$ into finite subset of Q_{1h}. Similarly we define δ_2 and δ_2'.

Define $\mathsf{A}_T = (\{r,u,\ell,d\}, q_0^T, \delta_T, q, F_T)$ is a finite deterministic automaton such that $L(\mathsf{A}_T) = T$ and the FSMPA , $\mathsf{A} = \left(\Sigma \cup \{\Diamond\}, \Gamma, Q_h, \delta, \delta', S_h, F_h, F_h, \$\right)$ such that $L(\mathsf{A}) = L_1 Ш_T L_2$. Here Q_h is the set of states associated with $\Diamond$, $Q_h = \bar{R}_h \cup R_{1h} \cup \ldots \cup R_{kh}$, $R_{ih} \neq R_{jh}$, $i \neq j$. Each R_{ih} has astarting state q_i or qh_i and a final state f_i or f_{hi}, $i = 1,2,\cdots,k$. $S_h = \bigcup_{i=1}^{k} \{q_i \cup q_{hi}\}$, $\bar{R}_h$ has a start state q_0. $F_h \subseteq \bar{R}$ is the end state. \$ is the end marker. Define $Q_h = Q_{1h} \times Q_T \times Q_{2h}$, $S_h = \{(q_0^1, q_0^T, q_0^2)\}$, or more precisely we define S_h as $\left\{q_{h10}(q_{h10}), q_0^T, q_{20}(q_{h20})\right\}$, $F_h = F_{1h} \times F_T \times F_{2h}$ and $F_h' = F_{1h}' \times F_T' \times F_{2h}'$ A run of A on an input $\mathsf{P} \in (\Sigma \cup \{\Diamond\})^{**}$ imitate either A_1 or A_2 and as time varies , changes the functioning from A_1 to A_2 or from A_2 to A_1. For each of the changes the transition in A_T as follows:

If change is from A_1 to A_2 then it is interpreted as u or d andit is r or ℓ respectively if it is from A_2 to A_1. The input P is accepted by A iff L_1, L_2 and T are recongnized byeach of $\mathsf{A}_1, \mathsf{A}_2$ and A_T respectively.

The mapping δ is defined as:

If T is a column shuffle, then the transition is

$$\begin{aligned}\delta((q_1,q_T,q_2),a) = \{&[(\delta_1(q_1,a),\delta_T(q_T,r),q_2)],[(\delta_1(q_1,\Diamond),\delta_T(q_T,r),q_2)],\\ &[(\delta_1(q_1,a),\delta_T(q_T,u),q_2)],[(\delta_1(q_1,\Diamond),\delta_T(q_T,u),q_{h2})],\\ &[(q_1,\delta_T(q_T,u),\delta_2(q_2,a))],[(q_1,\delta_T(q_T,u),\delta_2(q_2,\Diamond))],\\ &[(q_{h1},\delta_T(q_T,u),\delta_2(q_2,a))],[(q_{h1},\delta_T(q_T,u),\delta_2(q_2,\Diamond))]\}\end{aligned}$$

.If T is a row shuffle, then it is defined as

$$\begin{aligned}\delta((q_1,q_T,q_2),a) = \{&[(\delta_1(q_1,a),\delta_T(q_T,l),q_2)],[(\delta_1(q_1,\Diamond),\delta_T(q_T,l),q_2)],\\ &[(\delta_1(q_1,\Diamond),\delta_T(q_T,l),q_{h_2})],[(\delta_1(q_1,a),\delta_T(q_T,l),q_{h_2})],\\ &[(q_1,\delta_T(q_T,d),\delta_2(q_2,a))],[(q_1,\delta_T(q_T,d),\delta_2(q_2,\Diamond))],\\ &[(q_{h1},\delta_T(q_T,d),\delta_2(q_2,a))],[(q_{h1},\delta_T(q_2,d),\delta_2(q_2,\Diamond))]\}\end{aligned}$$

where $\{q_1, q_{h1}\} \in Q_{1h}$, $\{q_2, q_{h2}\} \in Q_{2h}$, $q_T \in Q_T$ and $\{r,u,l,d\} \in T$, δ' is defined from $\bar{R}_h \times \Gamma$ into $\bar{R}_h$. Thus we have $L(\mathsf{A}) = L_1 Ш_T L_2$ and $L_1 Ш_T L_2$ is a regular partial matrix array language.

Theorem. 2

For any two recognizable partial array languages L_1, L_2, $L_1 Ш_T L_2$ is a recognizable partial array language, where $T \subseteq \{r,u\}^* \cup \{l,d\}^*$ is regular.

Proof. Let L_1 and L_2 be the two recognizable partial array languages accepted by OTPA, A_1 and A_2 are such that $L(\mathsf{A}_i) = L_i, i = 1,2.$

Define $\mathsf{A}_i = \left(\Sigma \cup \{\Diamond\}, Q_i, Q_h^i, q_0^i, F_i, \delta_i\right)$, where Q_i is set of states, Q_h^i is the finite states associated with $\Diamond, q_o^i$ is the initial state and $q_0^i \in Q_i$, for $i = 1,2.$ $F_i \subseteq Q_i \cup Q_h^i$ are all final states. The transition function δ_i is defined as $\delta_i : Q_p \times Q_p \times (\Sigma \cup \{\Diamond\}) \to Q_p$.

The transition in A_T is as follows: a change from A_1 to A_2 (A_2 to A_1) is taken as u or d (r or l ,) respectively. The input P is accepted by A iff each of $\mathsf{A}_i, i = 1,2$ and A_T accepts L_1 and L_2 and T respectively.

Define $Q = Q_1 \times Q_T \times Q_2$, $Q_0 = \{q_0^1, q_T, q_o^2\}$, where $Q_1 = Q_1 \cup Q_h^1$ and $Q_2 = Q_2 \cup Q_h^2$, thus $Q = (Q_1 \cup Q_h^1) \times Q_1 \times (Q_2 \cup Q_h^2)$, $F = F_1 \times F_T \times F_2$.

The transition δ is defined as $\delta : Q \times Q \times (\Sigma \cup \{\Diamond\}) \to 2^Q$ and is given as follows:

$$\delta[(q_1^{'}, q_T, q_1^2), (q_2^1, q_T, q_2^2), b]$$
$$= \{[\delta_1(q_1^1, q_2^1, b), \delta_T((q_T, r), q_1^2)], [q_1^1, \delta_T(q_T, u), \delta_2(q_1^2, q_2^2, b)],$$
$$[\delta_1(q_1^1, q_2^1, b), \delta_T(q_T, l), q_1^2]\ , [q_1^1, \delta_T(q_T, d), \delta_2(q_1^2, q_2^2, b)]\}$$

and

$$\delta[(q_1^1, q_T, q_1^2), (q_2^1, q_T, q_2^2), \Diamond]$$
$$= \{[\delta_1(q_1^1, q_2^1, \Diamond), \delta_T((q_T, r), q_1^2)], [q_1^1, \delta_T(q_T, u), \delta_2(q_1^2, q_2^2, \Diamond)],$$
$$[\delta_1(q_1^1, q_2^2, \Diamond), \delta_T(q_T, l), q_1^2]\ , [q_1^1, \delta_T(q_T, d), \delta_2(q_1^2, q_2^2, \Diamond)]\}$$

where $q_1^1, q_2^1 \in Q_1, q_T \in Q_T, q_1^2, q_2^2 \in Q_2, b \in \Sigma$. By the above transition it can be easily verified that $L(\mathsf{A}) = L_1 Ш_T L_2$ and hence $L_1 Ш_T L_2$ is also recognizable.

4 Conclusion

The impact of shuffle on trajectories over a regular matrix partial array languages and on the recognizable partial array languages are studied. The procedure can be applied in automation and in recognition of certain classes of designs. This can be applied in understanding the transformation of shapes in Architectural application and in evolution of two dimensional patterns.

References

[1] J. Berstel and L. Boassan, *Partial words and a theorem of Fine and wilf*, Theoretical Computer Science, 218 (1999), 135-141.

[2] H. Geetha, D.G. Thomas, T. Kalyani and A.S. Prasanna Venkatesan, *Shuffle on trajectories over finite array languages*, in J.K. Aggarwal et.al (eds), International Conference on Combinatorial Image Analysis, Lecture Notes in Computer Science 6636, 2011, 261-274.

[3] H.Geetha, K.Sasikala, P.Venkatesh Babu and T.Kalyani *Shuffle on trajectories Over Partial Array Languages* presented in ICMTA 2020 and published in AIP Conference proceedings,volume2277,06November 2020 030001-1 - 030001-8.

[4] P. Helen Chandra, C.Martin-Vide, K.G. Subramanian, D.L. Van and P.S.P.Wang, parallel contextual array grammars and trajectories in Handbook of pattern recognition and computer vision.

[5] A. Mateescu, G. Rozenberg and A. Salomaa, *Shuffle on trajectories: syntactic constraints*, Theoretical Computer Science, 197 (1998), 1-56.

[6] D.K. Sheena Christy, V. Masilamani, D.G. Thomas, Atulya K. Nayar and Robinson Thamburaj, *Shuffle on array languages generated by array grammars* Mathematics for application, math.Appl. 3(2014), 17-31.

[7] F. Sweety, D.G. Thomas, V.R. Dare and T. Kalyani, *Recognizability of partial array languages*, The Journal of combinatorial Mathematics and Computational Computing (2007), 237-249.