JCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Special Issue, 5, March 2019 E-ISSN: 2347-2693

Generation of Certain Patterns Using Array-Token Petri Nets

V. Sharon Philomena[1*], P. Usha[2], R. Santhiya[3]

[1,3] PG Department of Mathematics, Women's Christian College, Chennai-6
[2]Department of Mathematics, D.G. Vaishnav College, Chennai-106

[*]*Corresponding Author: sharonphilo2008@gmail.com, Tel.:* +00-80156-67977

DOI: https://doi.org/10.26438/ijcse/v7si5.2529 | Available online at: www.ijcseonline.org

Abstract— A Petri net is a specific type of Mathematical modeling which is useful in data analysis, simulations, business process modeling and other such scenarios. There are different types of Petri net models which models the behaviour of distributed systems. This paper presents some applications of array token Petri nets by generating certain pattern of picture languages using evolutionary rules on transitions. An Array-token Petri net (ATPN) was first defined by Beulah Immanuel by labeling tokens by arrays which generates picture languages. This paper contains two rules called elementary evolution rule (EER) and parallel evolution rule (PER) which is used to generate same pattern of picture languages in different sizes. Application of ATPN which generates English alphabetic letters treated as rectangular arrays was examined and also Petri net which generates kolam pattern was also studied. Motivated by these papers, we have generated certain patterns of picture languages using ATPNs by applying some evolution rules on transitions.

Keywords**—** Petri net, Array-token Petri net, Picture languages.

I. INTRODUCTION

The concept of Petri net was introduced by Carl Adam Petri in 1962. A Petri net is one of several mathematical tools used to model the functionality or behavior of a distributed system with concurrency. It is a bipartite graph which has place nodes, transition nodes and directed arcs which connects places with transitions. The place from which the arc enters a transition is called the input place of the transition and the place to which an arc enters from a transition is called the output place of the transition. Places may hold any number of tokens. The tokens circulate the system between places via transitions. Distribution of tokens over the places of a net is called a marking. Transitions act on input tokens by a process known as firing. A transition triggers whenever there is at least one token in all the input places. When the transition fires, tokens are removed from its input places and added at all output places of the transition.

Syntactic methods of generation and recognition of patterns and pictures have been developed for many years by researchers with different motivations and have been applied in practical problems such as character recognition, two-dimensional mathematical symbols, 3D object recognition and many others. Several two-dimensional grammars which constitute one such area of syntactic methods, have been proposed and studied in [7].

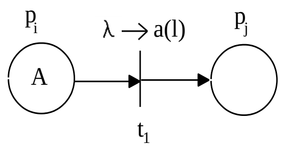
A coloured Petri net (CPN) has a transition which can fire with respect to its colours [3]. An extension of this coloured Petri net called string-token Petri net was introduced in [1] by labeling the tokens with strings and the transitions with evolution rules. On the other hand, an extension of the string-token Petri net called array-token Petri net was introduced in [4] by labeling tokens by arrays and is used to generate picture languages. A Petri net model to generate English alphabetic letters using array-token Petri net was examined in [9]. Petri net model to generate array of tiles and kolam patterns was studied in [5, 6]. In this paper, we have generated certain patterns of picture languages using ATPNs by applying some evolution rules on transitions.

II. PRELIMNARIES

Definition 2.1
An Array-token Petri net (ATPN) is a 6-tuple $N = (P, T, C, A, R, M_0)$ where
(i) P is a set of places;
(ii) T is a set of transitions;
(iii) C is set of symbols (colours) and C_{AY} is the set of all rectangular arrays over this colour set C, that are associated with the tokens.
(iv) $A \subseteq (P \times T) \cup (T \times P)$ is a set of arcs;
(v) R(t) is the set of evolution rules associated with a transition t;
(vi) M_0, the initial marking, is a function defined on P such that, for $p \in P$, $M_0(p) \in [C_{AY}]_{MS}$.

Definition 2.2

An Elementary Evolution Rule (EER) [8] over V_{AY}, where V an alphabet, is one of the following:
(i) Identity, which keeps the array unaltered;
(ii) Column insertion $\lambda \to a$ (l/r, according as it is left or right);
(iii) Row insertion $\lambda \to a$ (u/d, according as it is up or down);
(iv) Column deletion: $a \to \lambda$ (l/r, according as it is left or right);
(v) Row deletion: $a \to \lambda$ (u/d, according as it is up or down);
(vi) Substitution: $a \to b$, where $a, b \in V$ and λ is the empty word.
The following subnets illustrate how column insertion rules are applied on the left and right of the array A respectively. The rule $\lambda \to a(l)$, inserts a column of a's to the left of the array A , the rule $\lambda \to a(r)$, inserts a column of a's to the right of the array A.



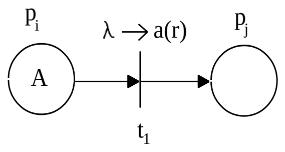


Figure. 1: Sub nets used for the column insertion

Similarly, the rule $\lambda \to a(u)$, inserts a row of a's to the top of the array A and the rule $\lambda \to a(d)$, inserts a row of a's to the bottom of the array A. Similar would be the case for the column deletion and row deletion. Substitution rule $a \to b$ just replaces a by b.

Definition 2.3
The language, which is generated by ATPN with EER on transition is denoted by AERL.

Definition 2.4
A Parallel Evolution Rule (PER) over V_{AY}, where V an alphabet, is one of the following:
(i) The rule $\lambda \to a(u, d, l, r)$ inserts 'a' simultaneously on up and down rows, left and right columns.
(ii) The rule $\lambda \to a(u, d);b(l, r)$ inserts 'a' on up and down rows; 'b' on left and right columns simultaneously.
(iii) Let c_1, c_2, c_3 and c_4 denote top left, top right, bottom left and bottom right corners of rectangular array respectively, the rule $\lambda \to a(c_1, c_2, c_3, c_4)$ inserts 'a' on top left, top right, bottom left and bottom corners of rectangular array simultaneously.
(iv) The rule $\lambda \to alt[a, b (u, d, l, r)]$ inserts a and b alternatively on up and down rows, left and right columns respectively.
The following subnets illustrate how insertion rule is applied on up, down, left and right simultaneously of a rectangular array A.

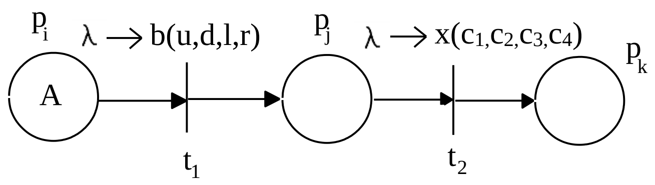


Figure. 2: Sub nets used for the array representation shown in Figure. 3(iii)

III. Results and Discussion

Theorem 3.1
There exist ATPNs with PER on transitions which generate certain patterns of picture languages.

Proof:
We construct ATPNs to generate certain picture languages obtained by pattern generation.
(i) Consider the following square tiles of same size as in Figure. 4. Using these tiles and parallel evolution rules (PER) on transitions of ATPN, let us generate a pattern called flower pattern as in Figure. 6.

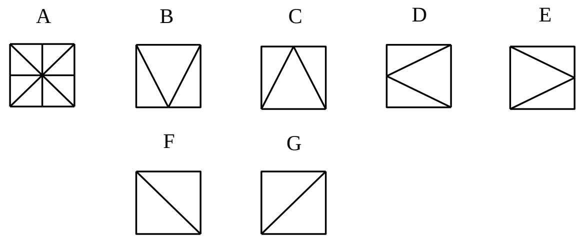


Figure. 4: Tiles used in ATPN$_1$

Construct the $ATPN_1$ as in the Figure. 5. In $ATPN_1$, when the transition t_1 fires it removes the token A from the input place p_1, performs the rule $\lambda \to [B(u);C(d);E(l);D(r)]$ on the array as indicated at the transition and deposits the array $\begin{matrix} & B & \\ E & A & D \\ & C & \end{matrix}$ in p_2 and again when the transition t_2 fires it performs the rule $\lambda \to [G(c_1,c_4);F(c_2,c_3)]$ and deposits the array $\begin{matrix} G & B & F \\ E & A & D \\ F & C & G \end{matrix}$ in p_3.
Now consider the array deposited in p_3 as P. When the transition t_3 fires it removes the token P from p_3, performs the rule $\lambda \to P(u, d, l, r)$ and deposits the array $\begin{matrix} & P & \\ P & P & P \\ & P & \end{matrix}$ in p_4 and at last when the transition t_4 fires, it performs the rule $\lambda \to P(c_1,c_2,c_3,c_4)$ and deposits the array $\begin{matrix} P & P & P \\ P & P & P \\ P & P & P \end{matrix}$ in p_3, we generate a pattern called flower pattern as in Figure. 6(iv). We can generate different sizes of this picture language as the number of times t_3 and t_4 fire.

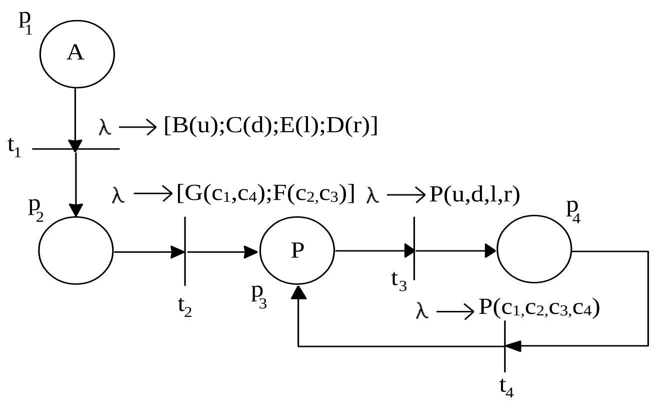
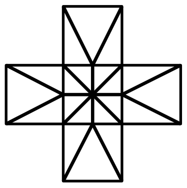
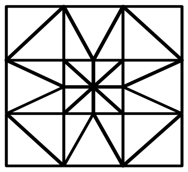
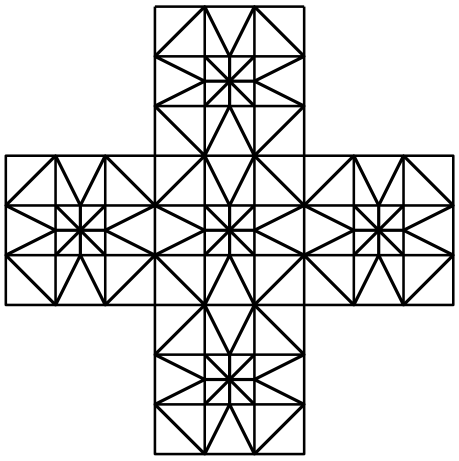


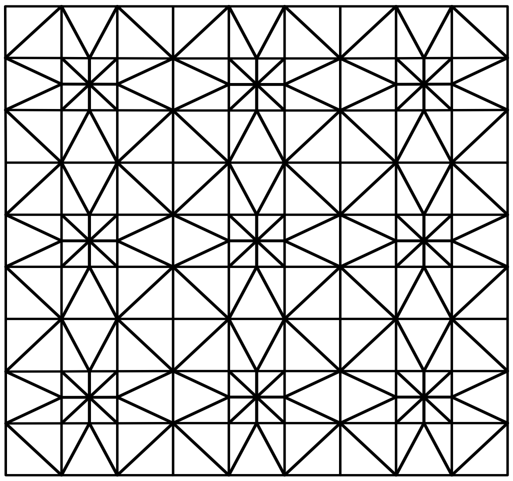
Figure. 5: ATPN$_1$

(i) When t_1 fires (ii) When t_1t_2 fires



(iii) When $t_1t_2t_3$ fires



(iv)When $t_1t_2t_3t_4$ fires

Figure. 6: Generation of flower pattern using ATPN$_1$

(ii) Consider the following square tiles of same size as in Figure. 7. Using these tiles and parallel evolution rules (PER) on transitions of ATPN, we can generate another pattern called swastik pattern which is given in Figure. 9.

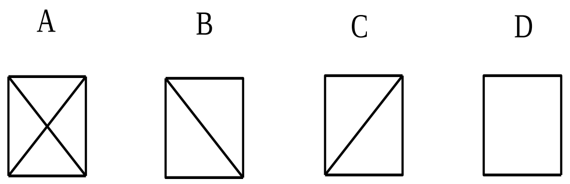


Figure. 7: Tiles used in ATPN$_2$

Construct the ATPN$_2$ as in the Figure. 8. In ATPN$_2$, when the transition t_1 fires it removes the token A from the input place p_1, performs the rule $\lambda \rightarrow [C(u, d);B(l, r)]$ on the array as indicated at the transition and deposits the array $\begin{array}{ccc} & C & \\ B & A & B \\ & C & \end{array}$ in p_2 and again when the transition t_2 fires it performs the rule $\lambda \rightarrow$ $[D(c_1,c_2,c_3,c_4)]$ and deposits the array $\begin{array}{ccc} D & C & D \\ B & A & B \\ D & C & D \end{array}$ in p_3. Now consider the array deposited in p_3 as P. When the transition t_3 fires it removes the token P from p_3, performs the rule $\lambda \rightarrow P(u, d, l, r)$ and deposits the array $\begin{array}{ccc} & P & \\ P & P & P \\ & P & \end{array}$ in p_4 and at last when the transition t_4 fires, it performs the rule $\lambda \rightarrow P(c_1,c_2,c_3,c_4)$ and deposits the array $\begin{array}{ccc} P & P & P \\ P & P & P \\ P & P & P \end{array}$ in p_3, we generate a pattern called swastik pattern as in Figure. 9. We can generate different sizes of this picture language as the number of times t_3 and t_4 fire.

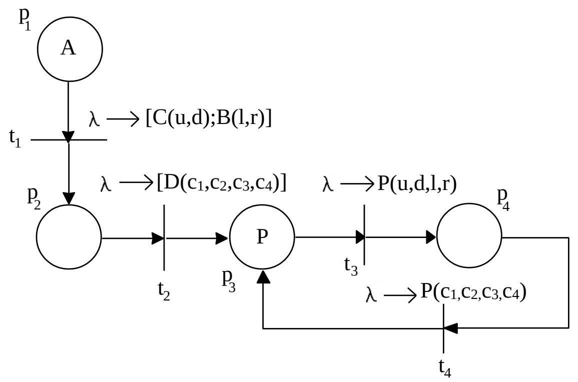


Figure. 8: ATPN$_2$

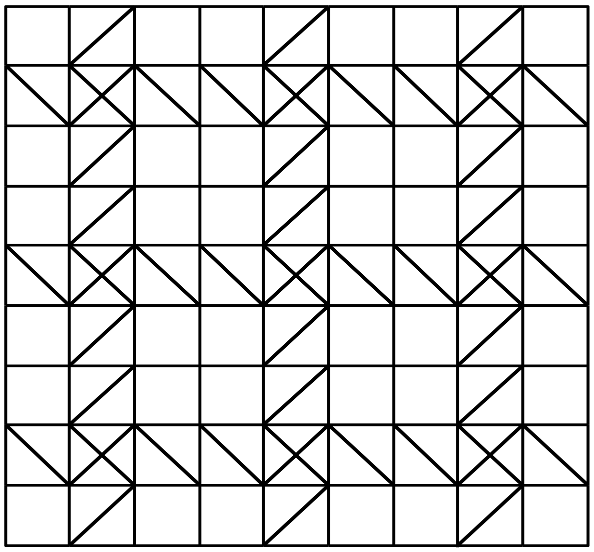


Figure. 9: Generation of swastik pattern using ATPN$_2$

(iii) Here we generate another swastik pattern using the same PER and the tiles as defined in Figure. 7.

Construct ATPN$_3$ as in Figure. 10. When the transition t_1 fires it removes the tokens A, B from p_1, performs the rule $\lambda \rightarrow$alt[B, A (r)] and deposits the array $B\ \ A\ \ B\ \ A\ \ B\ \ A\ \ B$ in p_1 and when the transition t_2 fires it performs the rule $\lambda \rightarrow$alt[D,C(u, d)] and deposits the array $\begin{matrix} D & C & D & C & D & C & D \\ B & A & B & A & B & A & B \\ D & C & D & C & D & C & D \end{matrix}$ in p_2 and when the transition t_3 fires it performs the rule $\lambda \rightarrow$alt[B, A(u, d)] and deposits the array $\begin{matrix} B & A & B & A & B & A & B \\ D & C & D & C & D & C & D \\ B & A & B & A & B & A & B \\ D & C & D & C & D & C & D \\ B & A & B & A & B & A & B \end{matrix}$ in p_3, we generate a pattern called swastik pattern as in Figure. 10. We can generate different sizes of this picture language as the number of times t_2 and t_3 fire.

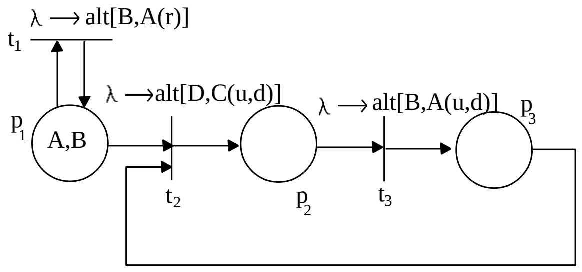


Figure. 10: ATPN$_3$

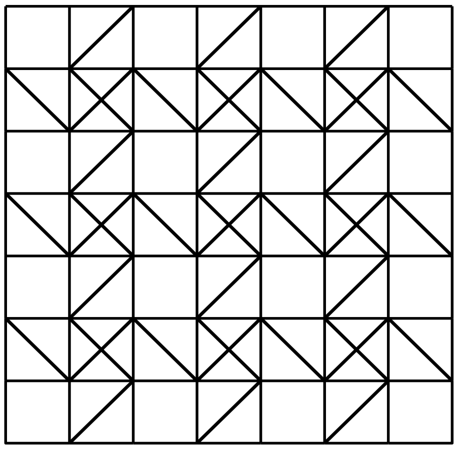


Figure. 11: Generation of another swastik pattern using ATPN$_3$

IV. CONCLUSION

In this paper, we have generated certain patterns of picture languages using some evolution rules on transitions of ATPNs. Also it is observed that using the above ATPNs the same patterns of picture languages can be generated in different sizes.

REFERENCES

[1] B. Immanuel, K. Rangarajan, K.G. Subramanian, "*String token-Petri nets*", Proceedings of the European Conference on Artificial Intelligence, One day workshop on Symbolic Networks, at Vanlencia, Spain, 2004.

[2] B. Immanuel, K.G. Subramanian, P. Usha, "*Array token petri nets and character Generation*", Proceedings of National Conference on Computational Mathematics and Soft Computing, Women's Christian College, pp. 68-72, 2009.

[3] K. Jensen, "*Coloured Petri nets*", Lecture Notes in Computer Science, 254, 1987 (248-299).

[4] S. Kannamma, K. Rangarajan, D.G. Thomas, N.G. David, "*Array token Petri nets, Computing and Mathematical Modeling*", Narosa Publishing House, New Delhi, India, pp. 299-306, 2006.

[5] D. Lalitha and K. Rangarajan, "*Characterisation of Pasting System using Array Token Petri Nets*", International Journal of Pure and Applied Mathematics, Vol. 70, No.3, pp. 275-284, 2011.

[6] D. Lalitha and K. Rangarajan, "*Petri nets generating Kolam Patterns*", Indian Journal of Computer Science and Engineering, Vol. 3, No.1, pp. 68-74, 2012.

[7] A. Rosenfeld and R. Siromoney, *"Picture languages – a survey, languages of design",* Vol. 1, pp. 229-245, 1993.

[8] G. Siromoney, R. Siromoney and Kamala Krithivasan, "Abstract families of Matrices and Picture Languages", Computer Graphics and Image Processing 1, pp. 282-307, 1972.

[9] P. Usha, B. Immanuel, R. Sattanathan, "*Application of Array token Petri nets in generating English Alphabetic letters*", The Journal of Combinatorial Mathematics and Combinatorial Computing, Vol. 79, pp. 91-98, 2011.

Authors Profile

Mrs.V. Sharon Philomena pursed Bachelor of Science from University of Madras, Chennai in 2002 and Master of Science from University of Madras in year 2004. She is currently pursuing Ph.D in Graph labeling. and working as Assistant Professor in PG Department of Mathematics, University of Madras, Chennai since 2010. She is a life member of Anna Periyar All India Mathematical Society. She has published more than 15 research papers in reputed International Journals including IJAER, International Journal of Computing algorithms. Organized conferences including UGC sponsored National workshops. She has received UGC –MRP a grant of 5 lakhs on Graph matching towards Women related Cancer. She has 15 years of teaching experience .

R. Santhiya pursed Bachelor of Science from Thiruvalluvar University, Vellore in 2017 and currently pursuing Master of Science from University of Madras, Chennai.