Ekoloji 28(107): 767-777 (2019)

An Intelligent Solution for a Sustainable Environment: Iso-Array Rewriting P Systems and Triangular Array Token Petri Net

K. Bhuvaneswari [1*], T. Kalyani [2], D. Lalitha [1]

[1] Department of Mathematics, Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu, INDIA

[2] Department of Mathematics, St. Joseph's Institute of Technology, Chennai, Tamil Nadu, INDIA

[*] Corresponding author: bhuvanamageshwaran@gmail.com

Abstract

Computational Sustainability is a nascent and growing field of computing that is concerned with the application of computer science principles, methods, and tools to problems of environmental and societal sustainability. This is not a one-way street, however, because sustainability problems force computer scientists into new theory, as well as new practice. In study of theory of computation, any computational model is always tested for its efficiency. A new computing model called P system is the one such efficient model which was introduced by Gh.Păunto generate string languages. The framework of the P system is like living cells in a biological system. Array rewriting P system is a model among variants of P system in which arrays are rewritten by the rules of array grammars. Conditional communication is a technique used for communication. Petri net is a model is used for dynamic systems. Array token Petri nets are the models which can generate array languages. In this paper parallel iso-array rewriting P systems are introduced with examples. The generating power of these P systems is examined with the generating power of some existing P system models. Also the computational power of these P systems is compared with the computational power of an existing petri net called triangular array token petri net.

Keywords: sustainable environment, sustainability, iso-triangular tiles, iso-array grammars, membrane computing, rewriting P system, permitting and forbidding conditions, Petri Nets

Bhuvaneswari K, Kalyani T, Lalitha D (2019) An Intelligent Solution for a Sustainable Environment: Iso-Array Rewriting P Systems and Triangular Array Token Petri Net. Ekoloji 28(107): 767-777.

INTRODUCTION

Sustainability can be taught at many points in the computer science curriculum. The following statements point to material that can help instructors infuse sustainability into the computer science curriculum. These materials range from entire courses dedicated to computing and sustainability, to stand alone exercises that contextualize a CS problem within a sustainability application. Areas of computing that are relevant to sustainability include:

- Computer Hardware and Architecture: Computing is ubiquitous in much of the world, and computing's collective energy footprint worldwide is growing. There is a great need for energy-efficient computer systems, from individual systems like a laptop to larger data centers of the "cloud", which are made from easily recyclable components that do not pollute the environment and threaten human health.

- Robotics and Sensors: Computing is increasingly used to monitor the natural and built environments, ranging from sensors that monitor civil infrastructure like bridges and the power grid, to sensors in-the-wild that identify species based on vision and audio inputs. These sensing capabilities include mobile platforms, most notably robots that are terrestrial, aquatic, or aerial, which are used in disaster response (e.g., oil spills and earthquake rescue) and routine monitoring of oceans, lakes, and savannahs, to name but a few.

- Cyber-Physical Systems: Beyond simpler forms of sensing, computing is increasingly embedded in and controls physical systems, such as cars, highways, and buildings; so as to improve energy efficiency (e.g., fuel consumption) and safety. Increasingly, human-built systems such as cars and airplanes are themselves robotic systems.

 Received: 25 Sep 2017 / Accepted: 26 May 2018

- Intelligent Systems: The capabilities for sensing and control above can only be realized through intelligent software, to include artificial intelligence (AI). Additionally, AI is vital for deliberative human decision making, such as resource planning (e.g., wildlife reserves, water usage), with optimization and machine learning being important supporting methods.

- Social Computing and Networking: Information and computing technology connects people, and thereby provides avenues by which social ties can change behavior, be it related to human health (e.g., quitting smoking, diabetes management), or environmental sustainability. Applications in sustainability would include forming online recycling cooperatives, and promoting purchase of ecologically friendly products based on a full life-cycle analysis.

- Mobile Computing: This technology is at the intersection of sensors, intelligent systems, social computing, and other areas, with human-carried mobile smart phones and cameras recording, transmitting, and analyzing data ranging from plants and animals, to consumer product bar codes. Other activities with these devices involve route and activity planning, all with implications for sustainability.

Inspired by the behavior and structure of living cells in a biological system, membrane computing called P system is a theoretical model, introduced by Gh. Paun (2002). In this P system membranes are named by the numerical numbers in 1-1 manner. In each region of each membrane rules and multi set of objects are present. Rules present in the regions corresponding to the membranes will do the computation. During the computation objects present in all regions are processed one by one by the rules in the regions in a non-deterministic and maximally parallel or sequential manner. At each step in the computation all objects can be evolved by the given rules only. The evolved objects can then be communicated toother regions, with the help of target indicators. If the target indicator is *here,* then the object is retained in the same membrane. If the target indicator is *in* then the computed string or array is entered in to the next immediate membrane. The target indicator *out* will send out the pattern to the outer region. If there is no rule to apply to the resulting picture, then the computation halts and the computation is successfully completed (Nagoba et al. 2017).

Among types of P system, rewriting P system is a model was introduced for strings and it has been investigated extensively by connecting Chomsky grammars. In array rewriting P system, grammar rules are applied to rewrite the strings. Parallel array rewriting P system is the special feature of rewriting P system and rewriting has been done in parallel manner. Besozzi et al. (2004) introduced parallel rewriting P system for string languages. In which rewriting of strings have been done in parallel mode. Rewriting P system with conditional communication has been studied by Bottoni et al. (2002) for strings. Motivated by rewriting P system for strings Ceterchi et al. (2003) introduced an array rewriting P system to compute two dimensional picture languages. To fill the floor different shapes of tiling patterns are required. Kalyani et al. (2006) proposed iso-triangular tiles which are gluable only if the edges of two different labeled tiles have the same length. Among variants of rewriting P system iso-array rewriting P system with context-free array rules is exclusively studied in Bhuvaneswari et al. (2016b). Contextual iso-array grammar rules have been used in contextual iso-array P system to generate triangular picture languages (Bhuvaneswari 2015, Bhuvaneswari et al. 2016a). Extending the literature on array rewriting for strings Subramanian et al. (2014) introduced parallel array rewriting P system. Parallelarray rewritingP system with tables of parallel array rewriting rules of grammars is defined in Linqiang et al. (2016). It is noted that in order to rewrite the arrays, two-dimensional array grammar rules have been used in parallel mechanism. To generate an array using such models, the number of membranes required is the least. In Subramanian et al. (2007) parallel array rewriting P system with unique parallel mechanism and conditional communication is considered (Çiftci 2016).

Petri nets (Mitra et al. 2014) are the models in mathematics proposed to dynamic system model. To simulate the activity of the dynamic system tokens are used and which are represented by black dots. The tokens move when the transition fires. Array token Petri nets (Lalitha 2015, Lalitha et al. 2012) are proposed to generate array languages. The arrays over an alphabet are used as tokens over an alphabet not the black dots. The transitions are associated with catenation rules. Firing of transitions helps to catenate the arrays to build bigger arrays. These models were also designed to generate patterns over triangular tiles. The tokens are triangular arrays in this model and which are made up of iso-triangular tiles. The generating power of triangular tile pasting P system and generating power of triangular array token Petri nets are investigated in Bhuvaneswari et al. (2016). In triangular array token

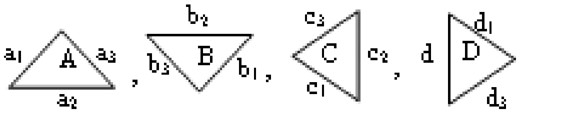
Petri net transitions associated with catenation rules explains the computation of the patterns. The firing rules in triangular array token petri nets are different from the firing rules of other Petri nets.

In this paper, section 2 recalls the basic definitions like iso-triangular tiles, pasting rules of iso-triangular tiles, iso-array grammars (RIAG, CFIAG). Also it recalls the basic Petri nets like Petri nets, token Petri nets, triangular array token Petri nets and catenation of triangular arrays. In section 3, parallel iso-array rewriting P system, parallel iso-array rewriting P system with tables of rules and parallel iso-array rewriting P system with conditional communication are introduced with suitable examples. Section 4 gives comparison results on the computational powers of the existing models of iso-arrays and parallel iso-array rewriting P systems. Also the comparison study on the generating powers of iso-array rewriting P systems and triangular array token Petri nets are given.

BASIC DEFINITIONS

Here we recollect the notion of triangular tiles and definitions of iso-array rewriting P system and array token petrinets. Basically a tile is a topological disc with closed boundary in the XOY plane, whose edges are gluable.

Definition 1 Consider the labeled triangular tiles,



,

whose horizontal and vertical edges are of unit length and other edges are of 1/√2 length.

Tiles A, B, C, D have the following pasting rules;

1. Tile A can glued with tile B by the pasting rules $\{(a_1, b_1), (a_2, b_2), (a_3, b_3)\}$, with tile C by the rule $\{(a_3, c_1)\}$ and with tile D by the rule $\{(a_1, d_3)\}$

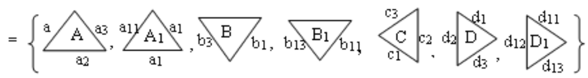
2. By the pasting rules $\{(b_1, a_1), (b_2, a_2), (b_3, a_3)\}$ tile B can be glue-able with tile Aand by the rule $\{(b_1, c_3)\}$ with tile C and by the rule $\{(b_3, d_1)\}$ with the tile D.

3. By the pasting rule $\{(c_1, a_3)\}$ tile C can be glued with tile A and by the pasting rule$\{(c_3, b_1)\}$ with tile B and with tile D by the pasting rules $\{(c_1, d_1), (c_2, d_2), (c_3, d_3)\}$.

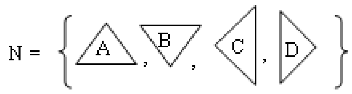
4. Tile D can be glued with tile A by the pasting rule $\{(d_3, a_1)\}$ with tile B by $\{(d_1, b_3)\}$ and with the tile C by the pasting rules $\{(d_1, c_1), (d_2, c_2), (d_3, c_3)\}$.

Notations:

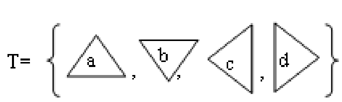
1. Iso-triangular tiles



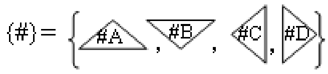
2. Non-terminal symbols



3. Terminal symbols

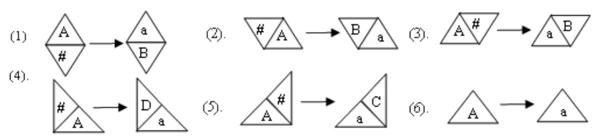


4. Empty symbols



Definition 2

A regular iso-array grammar (RIAG) is a structure G = (N, T, P, S) where N = {A, B, C, D}, T= {a, b, c, d} are finite sets of symbols (isosceles right angled triangular tiles) and N ∩ T =ϕ, Elements of N and T are called non-terminals and terminals respectively. S ϵ N, S is the start symbol of the axiom, P consists of rules of the following forms:



Similar rules can be given for the other tiles B, C and D. The set of all languages generated by RIAG is RIAL.

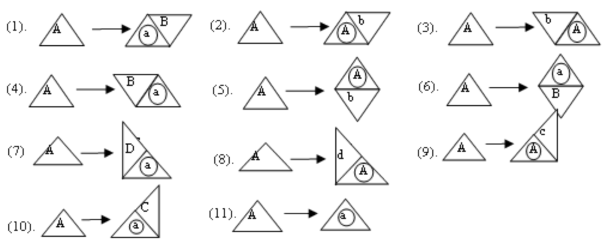
Definition 3

A Context-Free Iso-Array Grammar (CFIAG) is a structure G = (N, T, P, S) where N and T are finite nonempty set of symbols (isosceles right angled triangular tiles), N ∩ T = ϕ. Elements of N and T are called non-terminals and terminals, respectively. S ϵ N is the start symbol or the axiom. P consists of rules of the form $\alpha \rightarrow \beta$, where α and β are finite connected array of one or more iso triangular arrays over N ∪ T ∪ # and satisfy the following conditions.

1. The shapes of α and β are identical.
2. α contains exactly one non terminal and possibly one or more #s.
3. Terminals in α are not rewritten.
4. Application of the rule $\alpha \rightarrow \beta$ preserves the connectedness of the host array (that is, the application of the rule to a connected array results in a connected array).

The rule $\alpha \rightarrow \beta$ is applicable to a finite connected array γ over $N \cup T \cup \#$, if α is a sub array of γ and in a direct derivation step, one of the occurrences of α is replaced byβ, yielding a finite connected array δ. We write $\gamma \Rightarrow_G \delta$ is denoted by $\Rightarrow_G\star$. The array language generated by G is defined by $\{\delta$: $S \Rightarrow_G\star$, δ is a finite connected array over T$\}$ and is denoted byL(G).The class of iso-picture languages by CFIAG is denoted by £(CFIAG).

Definition 4. Context-free basic puzzle Iso-Array Grammar (CFBPIAG) is a structure G = (N, T, P, S), where N, T, and S are defined as in Definition 2.3 and P is the set of rewriting rules of the form $X \rightarrow \alpha$, where X can be anyone of the tiles and α is a finitely connected array of one or more iso-triangular arrays, each cell containing either a non-terminal or a terminal symbol, with the symbol in one of the cells of α being circled.



In a direct derivation step, a non-terminal X in a cell is replaced by the right side α of the rule $X \rightarrow \alpha$. In the replacement, the circled symbol occupies the cell with symbol X and the remaining symbols of α occupying their respective relative positions with respect to the circled symbol of α. Again the rewriting by $X \rightarrow \alpha$ is possible only when the cells to be filled by the non-circled symbols of α containing the blank symbols # the class of iso-picture language generated by CFPIAG is denoted by £ (CFPIAG).

Definition 5. An iso- array-rewriting P system of degree $m \geq 1$, using rules of context free-iso- array grammars can be defined as $\prod = (V, T, \#, \mu, F_1, F_2, \ldots F_m, R_1, R_2, \ldots, R_m, i_0)$, where V is the finite set of total alphabets (iso-triangular tiles) and $T \subset V$ is the finite set of terminal alphabets, #set contains blank symbols, μ is the membrane structure associated with m regions. F_1, F_2, . . . F_m are finite set of iso-arrays over V associated with m regions of μ. R_1, R_2, …. R_m are the finite sets of iso-array rewriting rules over V associated with the m regions of the membrane system. The rules of context free iso-array grammars are associated with the target indicators tar = "{here, out, in}". The target indicator "here" means that the resultant picture retained in the same region, "out" means the picture after evolution rules sent out of the current region, the target indication "in" means the resultant picture enter into the immediate inner region. i_o is the label of elementary membrane (output membrane) of the system. If there is no internal membrane, then the target indication "in" could not be applied.

The process or computation is said to be successful if the generated picture reach the output membrane by applying the rules of context free iso-array grammars. The result of a halting computation consists of iso-triangular arrays over T which is presented in the region of the outer membrane. The set of all such computed or generated triangular arrays by the system $\prod$ is denoted by IAL($\prod$). The family of all iso-array languages IAL($\prod$) generated by $\prod$ with m membranes is denoted by $IARP_m$ (CFIAG or CFPIAG).

Definition 6. A Petrinet structure is a four tuple C= (P,T,I,O) where $P=\{p_1,p_2,\ldots\ldots\ldots p_n\}$ is a finite set of places , $n \geq 0$, $T = \{t_1, t_2, \ldots\ldots, t_m\}$ is a finite set of transitions $m \geq 0$, $P \cap T = \emptyset$, $I:T \rightarrow P^\infty$ is the input function from transitions to bags of places and $O:T \rightarrow P^\infty$ is the output function from transitions to bags of places.

Definition 7. A Triangular ArrayToken PetriNet (TATPN) is a six tuple N = ($\sum$, C, μ, S, σ, F) where $\sum$ is an alphabet of tiles, C is a Petri net structure, μ is an initial marking of arrays made up of tiles kept in some places of the net, S is a set of catenation rules, σ is a partial mapping from T to a set of catenation rules S, F is a sub-set of P, the set of places of the Petri net, which is the final set of places. The array language generated by a TATPN is the collection of all arrays that reach the places of F the final set of places.

To explain the language generated: Starting with the initial marking of arrays in certain places of the net all possible firing sequences are considered. As the transitions fire the arrays grow in size and also move from one place to another. Only the arrays that reach the places belonging to F are collected as the language generated.

PARALLEL ISO-ARRAY REWRITING P SYSTEMS

In this section, variants of rewriting P system called parallel iso-array rewriting P system, parallel iso-array rewriting P system with tables and parallel iso-array rewriting P system with conditional communication are introduced with suitable examples. In this P systemiso-arrays are rewritten by parallel mechanism. It means that in a region, iso-arrays are rewritten only by iso-array grammar rules and which are associated with unique target symbols.

Formally an iso-array rewriting P system is constructed as follows.

Definition 8

A parallel iso-array rewriting P system is a construct $\prod = (V, T, \#, \mu, F_1, F_2, \ldots F_m, R_1, R_2, \ldots. R_m, i_0)$, where V is the finite set of total alphabets, $T \subset V$ is the finite set of terminal iso-arrays. # is the blank symbol, μ is the membrane structure over V associated with m regions. $R_i(1 \leq i \leq m)$ are finite sets of iso-array rewriting rules over V. The iso-array rewriting rules are either Regular or Context-free with the target symbols or indicators from tar = {here, in_j, out}.

The computational process starts in the region one with the initial iso-array and rewriting of iso-arrays have been made by the iso-array grammar rules. The process is stopped, if no further rewriting rule can be applied in the generated iso-arrays. Hence the computation halts at the region of the output membrane. Very important point is rewriting of iso-arrays can be done in parallel mechanism by the application of iso-array rewriting rules which are associated with same target indicators. Due to this feature to generate iso-arraysonly less number of membranes is enough.It is comparatively very less than the number of membranes in iso-array rewriting P System with iso-array rules. The set of all picture languages generated by P-IARP (∏) is denoted by L(P-IARP(∏)). The set of all picture languages is denoted by P-IARP_m (RIAG/C-FIAG).

Further application in iso- array rewriting is parallel iso-array rewriting rules are taken in tables to generate some iso-triangular arrays. A region may have two or more than two such table rules. Formally parallel iso-array rewriting P system with tables of iso-array rules is defined as follows.

Definition 9. The parallel iso-array rewriting P system with table of iso-array rules is a construct $\prod = (V, T, \#, \mu, F_1, F_2, \ldots F_m, R_1, R_2, \ldots. R_m, i_0)$ where V is a finite set of non-terminal and terminal iso-triangular tiles, Tis finite set of terminal iso-triangular tiles. μ is the membrane structure containing m labeled membranes. $F_i(1 \leq i \leq m)$ is a finite set of iso- arrays initially present in m regions over the finite set V. R_i is a finite set of iso- array rewriting rules(Context-free iso-array rules/ Regular iso-array rules)associated with a target indicator from target =“{here, out, in_j}”.i_0 is the output membrane which collects the successive member of the picture languages. To computing the triangular arrays, the tables of iso- array rewriting rules are applied. In each table two or more than one rewriting rules should be presented and which are associated with the target symbols. Every region should have two or more than two tables of rules. If only one table of rules is available in the P system in every region then the system become a parallel iso-array rewriting P system. In a region tables of rules are chosen non-deterministically.

Computation is started from the initial iso-triangular array present in the first region of the membrane structure. The tables of iso-array rewriting rules in the region can be applied to the initial triangular array in non-deterministic way. The generated iso-triangular array is sent to the region j $(1 \leq j \leq m)$ or sent out from the current region or it stays in the same region depends only on the target indicator associated with that table rules. This generating process is continued until the non-terminals are rewritten by the terminal iso-triangular arrays. If there is no rule can be applied to the generated iso-arrays and all non-terminal iso-triangular arrays are rewritten by the terminals then the computational process is stopped. That is halting triangular picture is the generated iso-triangular array whose labels are terminals and no further table of rules can be applied to the generated picture. The set of all picture language generated by iso-array rewriting P system with Parallel iso-array rewriting rules in table is denoted TP-IARP_m (CFIAG/CFBPIAG/RIAG).

Definition 10. An iso-array rewriting P system with conditional commnication is defined as $\prod = (V, T, \#, \mu, L_i, (R_i, P_i, F_i), i_0)$ where V is a finite set of total alphabets(non-terminal and terminal iso-triangular tiles) Tis finite set of terminal iso-triangular tiles. μ is the membrane structure labelled as 1, 2, 3, ..., m in one-one way $(1 \leq i \leq m)$. L_i is a finite set of iso- arrays over V initially present in m regions over the finite set V. R_i is a finite set of iso-array rewriting rules (Context-free iso-array rules/ Regular iso-array rules).P_i and F_i are permitting and forbidding conditions associated with a target indicator from target = “{out, in_j}”, which are present in the region i(1 ≤ i ≤ m). i_0 is the output membrane or skin membrane, it collects the generated iso-triangular arrays.

P_i's and F_i's are permitting and forbidding conditions present in the regions i(1 ≤ i ≤ m)which are of the forms empty or symbol checking or sub-iso-array checking.

Conditions are given below:

1. Empty: There is no restriction implied on iso-arrays; they either exit from the current membrane or enter into any of the membrane directly inner to the membrane freely; we denote an empty permitting condition by (true, α), α∈ target, and an empty forbidding condition by (false, not α), α∈target.

2. Symbols (iso-triangular tiles) checking: Each P_i is a set of pairs (x, α), α∈ target and x∈V. Each F_i is a set of pairs (y, not α), α∈ target and y∈V. Generated iso-array A∈V* in a region by applying the iso-array rewriting rules can go to the membrane j only if there is a pair (x, in_j)∈P_i with x∈A and for each (y, $notin_j$)∈ Fi, y∉A; similarly the generated iso-arraysent out from the membrane jif there is a pair (x, out) ∈$P_i$and (y, notout) ∈ Fi for all y∉A.

3. Sub iso-array checking: Each P_i is a set of pairs (x, α), α∈ target and x∈V^+. Each F_i is a set of pairs (y, not α), α∈ target and y∈V^+. Generated iso-array A in a region by applying the iso-array rewriting rules can go to the membrane j only if there is a pair (x, in_j)∈P_i with x∈A and for each (y, $notin_j$)∈ Fi, y∉ A; similarly the generated iso-arrays ent out from the membrane jif there is a pair (x, out) ∈ P_i and (y, notout) ∈ Fi for all y∉A.

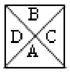
The working rule of this system: In ech region, iso-arrays are rewritten only by the rewriting iso-array rules. The resultant array is then communicated to other regions by the permitting and forbidding conditions present in the respective region. According to the conditions the iso-triangular arrays immediately enter into the membrane j(1 ≤ j≤ m) or sent out from the current membane i(1 ≤ i ≤ m). If the generated iso-array fulfils both permitting and forbidding conditions then the condition can be chosen non-deterministically. But if no conditions are fulfilled by the generated array then the iso-array is retained in the same region. The set of all iso-triangular arrays computed by an iso-array rewriting P system ∏ with conditional communications is denoted by P-$IARP_m$ (Cond, α, β) where α, β ∈ { empty symbol, symbol, sub iso-array}.

Example 1.

A class of picture language consists of rhombuses is generated by iso-array rewriting P system.

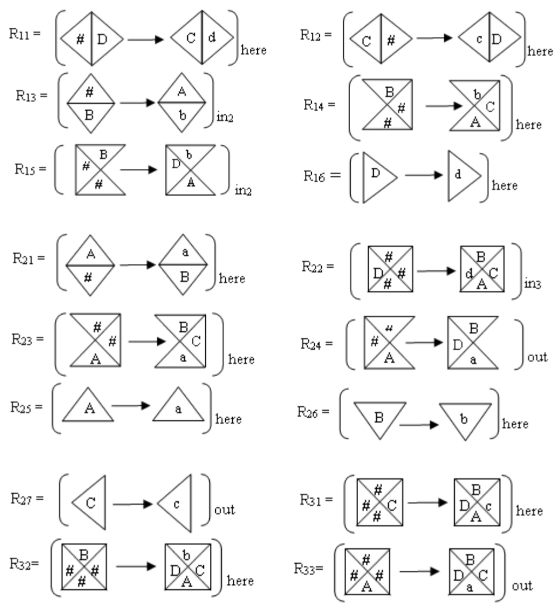
Consider the iso-triangular array rewriting P system with three membranes$\prod_1$= (V, T, #, $[_1[_2]_2\ [_3]_3]_1$, F_1, F_2, F_3, R_1, R_2,R_3,1) with the rules of context-free iso-array grammar.

Here N= {A, B, C, D}, T= {a, b, c, d},

F_1= 

, F_2 =ϕ, F_3=ϕ,

R_1= {R_{11}, R_{12},R_{13},R_{14},R_{15},R_{16}}, R_2 = {R_{21}, R_{22},R_{23}, R_{24},R_{25}, R_{26}, R_{27}}, R_3 = {R_{31}, R_{32},R_{33}}and region one is the output region.



The computation is stated in the first region, the rules R_{11}, R_{12} and R_{13} are applied to the axiom F_1. The target symbol in_2 associated with R_{13} send the picture into the region two. In two the rules R_{21}, R_{25}, R_{26}, R_{27} are applied. The target symbol associated with R_{27} send out the result picture to the region one again. In region one the rule R_{16} is applied to the existing iso-array and due the target symbol associated with the rule, the picture is haltered. The computation is successfully completed and the first member of the triangular array is collected in the output region one. To compute the second and third member of the picture language, the sequences of the rules R_{11}, R_{12}, R_{13}, R_{21}, R_{22}, R_{31}, R_{32}, R_{33}, R_{11}, R_{12}, R_{13}, R_{21}, R_{25}, R_{26}, R_{27}, R_{16} and R_{11}, R_{12}, R_{13}, R_{21}, R_{22}, R_{31}, R_{32}, R_{33}, R_{14}, R_{15}, R_{23}, R_{24}, R_{11}, R_{12}, R_{13}, R_{21}, R_{25}, R_{26}, R_{27}, R_{16} are applied successfully. The triangular picture language consists of Rhombuses are given in **Fig. 1**.

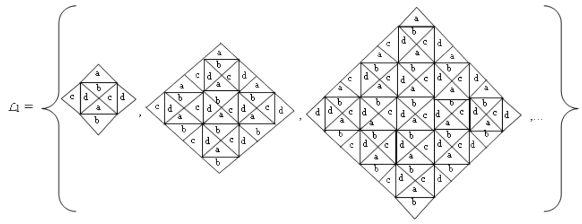


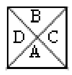
Fig. 1. A class of language consists of rhombuses

Example 2.

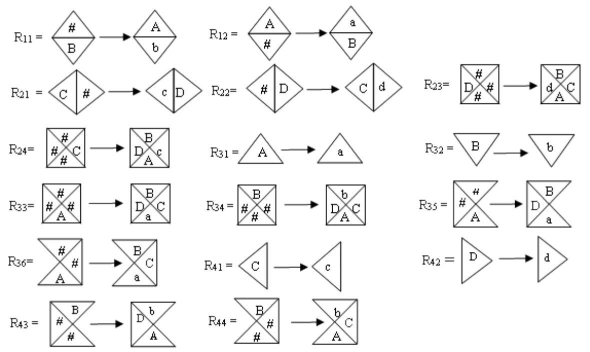
A class of triangular picture language consists of rhombuses is generated by the parallel iso-array rewriting P system with four membranes.

The parallel iso-array rewriting P system with context-free rules is

$\prod_1 = (V, T, \#, [_1[_2]_2\ [_3]_3[_4]_4]_1, F_1, F_2, F_3, F_4, R_1, R_2, R_3, R_4, 1)$ where V={A, B, C, D, a, b, c, d}

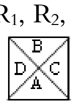
T={a, b, c, d}, $F_1=$  , $F_2=\phi$, $F_3=\phi$, $F_4=\phi$, $R_1=\{(R_{11}, in_2), (R_{12}, in_2)$,

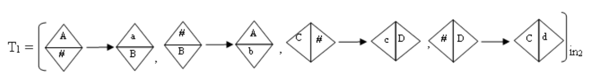
$R_2=\{(R_{21}, in_3), (R_{22}, in_3), (R_{23}, out), (R_{24}, out)\}$, $R_3=\{(R_{31}, in_4), (R_{32}, in_4), (R_{33}, in_2), (R_{34}, in_2), (R_{35}, in_4), (R_{36}, in_4)\}$, $R_4=\{(R_{41}, out), (R_{42}, out), (R_{43}, in_3), (R_{44}, in_3)\}$ where

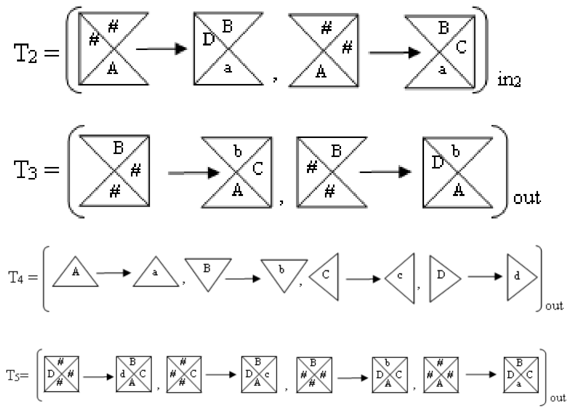


Example 3.

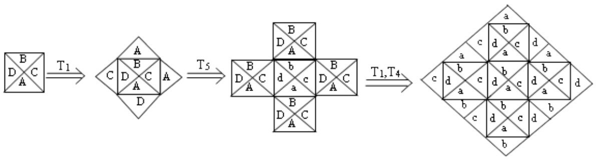
Consider the iso-array rewriting P system with tables of parallel context-free iso-array rules

$\prod_2 = (V, T, \#, [_1[_2]_2]_1, F_1, F_2, R_1, R_2, i_0)$, where N= {A, B, C, D}, T= {a, b, c, d}, $F_1=$  , $F_2=\phi$, $R_1=\{T_1, T_2\}$, $R_2=\{T_3, T_4, T_5\}$ and



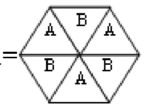


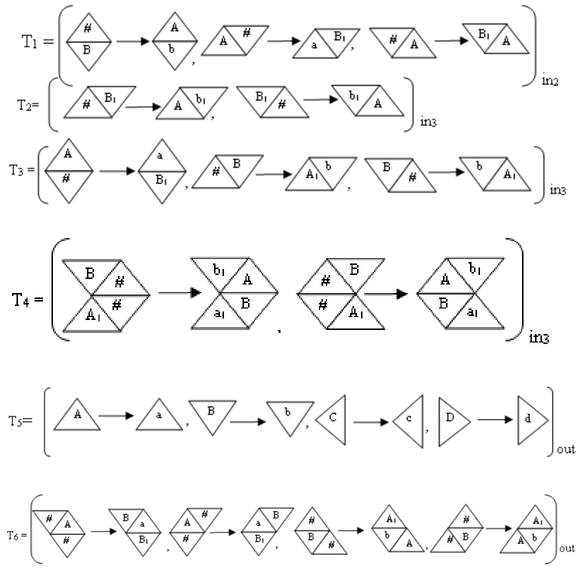
This P system generating a class of language consisting rhombuses. Second member of the language is shown with the derivation step below. The language consists of rhombuses are shown in **Fig. 1**.

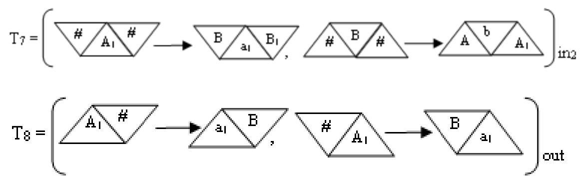


Example 4.

Consider an iso-array rewriting P system with table-parallel context-free iso-array rules

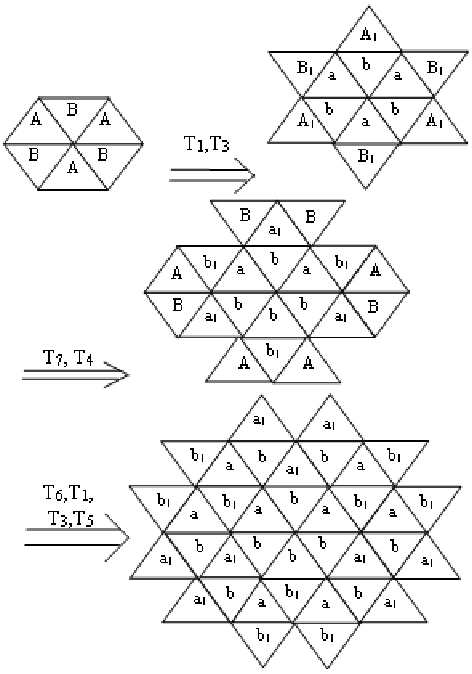
$\prod_3 = (V, T, \#, [_1[_2]_2[_3]_3]_1, F_1, F_2, F_3, R_1, R_2, R_3, i_0)$, where N={A, B, C, D}, T= {a, b, c, d}, $F_1=$  , ϕ, $R_1=\{T_1, T_2,\}$, $R_2=\{T_3, T_4\}$, $R_3=\{T_5, T_6, T_7, T_8\}$ and





This tables of parallel iso-array rewriting P system with context-free iso-array rules generating star picture language with equal arms on the six sides.

Derivation steps of second member of the picture language is shown below.



Theorem 1.

$IARP_m(RIAG) \subset IARP_m(CFIAG)$

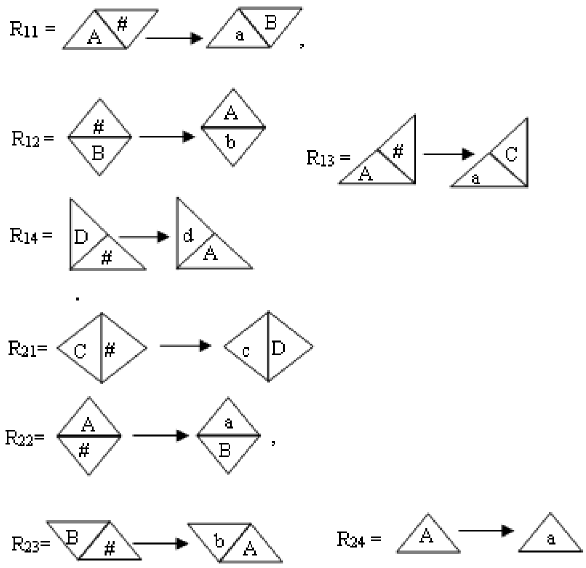
Proof:

The rewriting rules in regular iso-array grammar is contained in the rules of context-free iso-array grammar. A picture language generated by the rules of RIAG can also be generated by the rules of CFIAG But the picture language generated by the rules of CFIAG cannot be generated by the rules of RIAG. It can be explained with suitable examples. The iso-array rewriting P system $\prod_1$ with the rules of context-free iso-array grammar generated a picture language consists of arrow heads with the iso-triangular tiles c and d on the top.

$\prod_1 = (\{A, B, C, D\} \cup \{a, b, c, d\}, T, [_1[_2]_2]_1, F_1=A, F_2 = \phi, R_1, R_2, i_0=1)$

Here $R_1=\{(R_{11}, \text{here}), (R_{12}, \text{here}), (R_{13}, \text{in}), (R_{14}, \text{in})\}$,

$R_2=\{(R_{21}, \text{out}), (R_{22}, \text{here}), (R_{23}, \text{here}), (R_{24}, \text{here})\}$ and the CFIAG rules are given below.



The picture languages consists of arrow heads is shown in **Fig. 2**

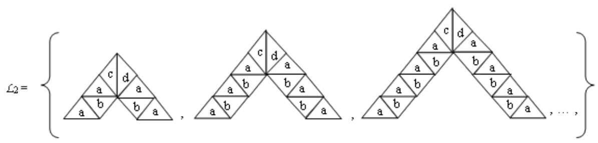


Fig. 2. A class of language consists of arrow heads and top tiles are c and d

and this class of picture language can also be generated by $IARP_m$ with the rules of RIAG. But the the class of language consists of arrow heads on the top iso-triangular tiles aand b generated by CFIAG shown in **Fig. 3** can not be generated by the rules of RIAG. Because in the rules of RIAG if the non-terminal tile is rewritten once by the terminal tile cannot be rewritten again.

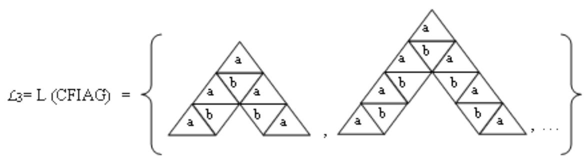


Fig. 3. A class of language consists of arrow heads and top tile is a

It proves that the language L_3 can not be generated by any RIAG.

Theorem 2.

L($PIARP_1$)=L($TPIARP_1$)

It is obvious from the definitions 8 and 9.

Theorem 3.

L($PIARP_m$)⊆L($TPIARP_m$)

Proof:

The picture language L_1 consists of rhombuses given in example 1 is generated by PIARP with four membranes (refer example 2). The language L_1 can also be generated by the tabled parallel iso-array rewriting P system with two membranes. And the language consists of star like pictures can be generated by parallel iso-array rewriting p system (refer example 4) with more than three membranes can not be generated by parallel iso-array rewriting P system. Hence the result is true.

Lemma 1.

Membranes required to generate a triangular picture language by $TPIARP_m$ is less than the number of membranes required by $PIARP_m$.

Proof: It is very clear from the example 2 and example 3.

Lemma 2.

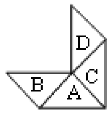
Membranes required to generate a triangular picture language by $TPIARP_m$ is less than the number of membranes required by $IARP_m$.

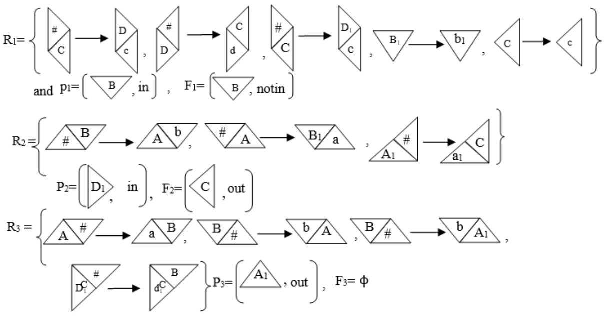
Proof: It is very clear from the example 1 and example 3.

RESULT

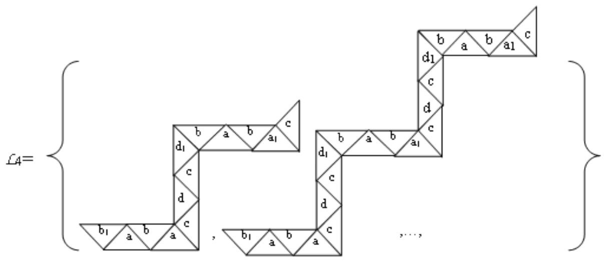
$PIARP_3$ (Cond, α, β) (RIAG) ⊆ $PIARP_3$ (Cond, α, β) (CFIAG)

Consider the parallel iso-array rewriting P system with conditional communication $\prod_4$=(V, T, #, $[_1[_2[_3]_3]_2]_1$, L_i, (R_i, P_i, F_i),i_0) where V contains the set of labelled non-terminal iso triangular tiles ={A, A_1, B, B_1, C, D, D_1} and T is the set of all labeled terminal iso triangular tiles {a, a_1, b,b_1,c,d,d_1}. L_i are the iso-triangular arrays present in the regions of the membranes 1, 2 and 3 respectively. R_i are the regular iso-triangular array grammar rules P_i are the permitting conditions and F_i are the forbidding conditions present in the regions i of the membranes i (1 ≤ i ≤ 3) and one is the output membrane.

Here L_1 =  and $L_2 = \emptyset, L_3 = \emptyset$, The iso-array rewriting rules are given below.



The picture language of stair case model is generated by the regular iso-array rewriting rules applied with the initial picture present in the first region.



Theorem 4.

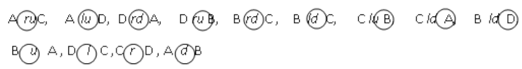
Parallel iso-array P system can not be comparable with the triangular array token petri nets.

Proof:

The class of picture language L_1 shown in **Fig. 1** is generated by Parallel iso-array rewriting P system can also be generated by TATPN. For, consider the TATPN which is the six tuple

N_1= (∑, C, μ,S, σ, F) where ∑ = { A, B, C, D},C=(P,T,I,O) , P={p_1,p_2,..........p_{12}},

T = {t_1, t_2, ..., t_{13}}. The initial marking μ is the array R, given in **Fig. 5**, in the place p_1. The set S of catenation rules are given below.



σ the mapping from the set of transitions to the set of rules is shown in the **Fig. 4** and F = { p_1}

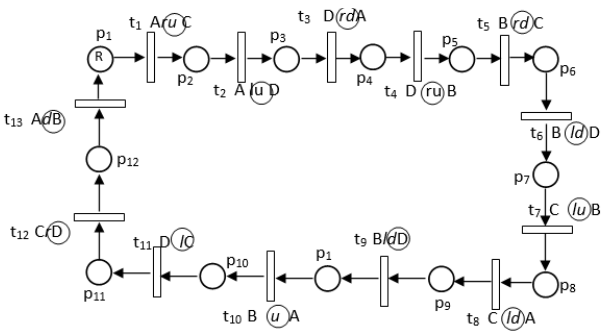


Fig. 4.

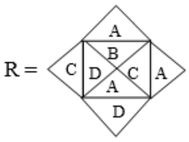


Fig. 5.

Starting with the array R, when every transition fires the corresponding tile is joined in parallel manner the given direction and the array is put in the output place. The result of firing the sequence of transitions $t_1t_2 \ldots t_9$ is given in **Fig. 6**. Firing the sequence $t_{10}t_{11}t_{12}t_{13}$ generates a bigger diamond, which is shown in **Fig. 7**. This array is the second pattern of the language generated by the net since it reaches the final place p_1.

The triangular picture language consists of star like shown example 4 cannot be generated by TATPN. But it can be generated by $PIARP_m$ (refer example 4).

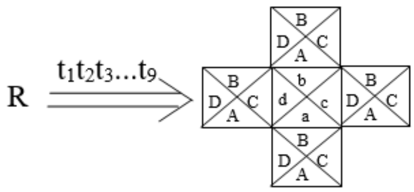


Fig. 6.

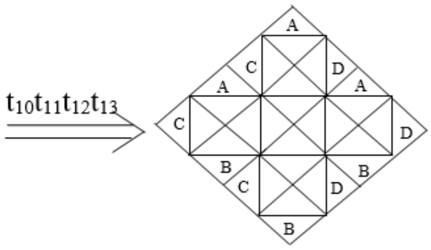


Fig. 7.

Theorem 5.

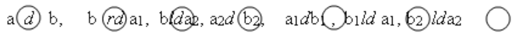
Triangular array token petrinets cannot be comparable with iso-array rewriting P system with regular iso-array rules.

Proof

The triangular picture language consists of arrow heads can be generated by triangular array token petrinets which is shown below.

Consider the TATPN which is the six tuple $N_2 = (\sum, C, \mu, S, \sigma, F)$ where $\sum = \{ a_1, a_2, b_1, b_2 \}$.

$C = (P,T,I,O)$, $P = \{p_1, p_2, \ldots, p_7\}$, $T = \{t_1, t_2, \ldots, t_7\}$. The initial marking μ is the array S is in the place p_1. The set S of catenation rules are given below.



σ the mapping from the set of transitions to the set of rules is shown in the **Fig. 8** and $F = \{ p_1 \}$.

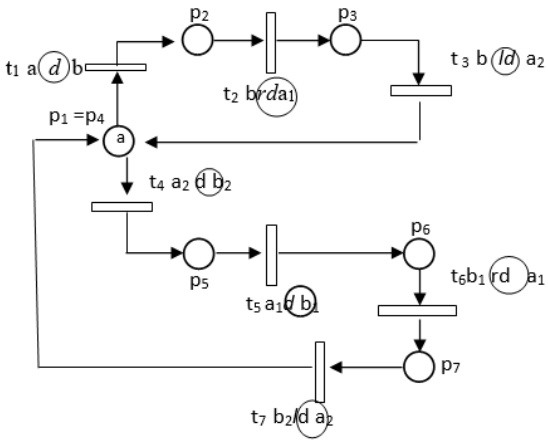


Fig. 8.

But this class of picture language consists of arrow heads with the tile a on the top cannot be generated by an iso-array rewriting P system with the rules of regular iso-array grammar. Since in regular array iso-array grammar rule once if a non-terminal iso-triangular tile is rewritten by a terminal iso-triangular tile then further the tile cannot be rewritten.

CONCLUSION

In this paper parallel iso-array rewriting P system and table parallel iso-array rewriting p system are introduced with examples. It is noted that to generate a triangular picture language by table parallel iso-array rewriting P system we required minimum number of membranes than requirement of membranes in parallel iso-array rewriting P system. The generating powers of parallel and table-parallel iso-array rewriting P systems are examined with the existing models called iso-array rewriting P system with iso-array rules and triangular array token petrinet. It is clear that iso-array rewriting P system with iso-array rules is not comparable with triangular array token petrinets.

REFERENCES

Besozzi D, Mauri G, Zandron C (2004) Hierarchies of parallel rewriting P systems -a survey. New Generation Computing, 22(4): 331-347.

Bhuvaneswari K (2015) Contextual Iso-Triangular Array P System Models. International journal of Applied Engineering Resesarch, 10: 5841-5849.

Bhuvaneswari K, Kalyani T, Gnanaraj Thomas D (2014a) Contextual Iso-Triangular Array P System and Iso-Triangular Picture Languages. Asian Conference on Membrane Computing, IEEE: 1-5.

Bhuvaneswari K, Kalyani T, Gnanaraj Thomas D, Nagar AK, Robinson T (2014b) Iso-array rewriting P Systems with Context-Free Rules. Mathemataics for Applications, 3(1): 1–16.

Bhuvaneswari K, Kalyani T, Lalitha D (2016) Triangular Tile Pasting P system and Array Generating Petrinets. International Journal of Pure and Applied Mathematics, 107(1): 111-128.

Bottoni P, Labella A, Martin-Vide C, Păun Gh (2002) Rewriting P Systems with Conditional Communication, Lecture Notes in Computer Science, Springer, Berlin, pp. 325–353, 2300.

Ceterchi R, Mutyam M, Păun Gh, Subramanian KG (2003) Array-rewriting P systems. Natural Computing, 2: 229-249.

Çiftci S (2016) Letter to Editor: Prognostic Value of Mean Platelet Volume and Platelet to Lymphocyte Ratio in Laryngeal Carcinoma. J Clin Exp Invest, 7(4):294-5. doi: 10.5799/jcei.328537

Kalyani T, Sasikala K, Dare VR, Robinson T (2006) Triangular Pasting System, In: K.G. Subramanian, K. Rangarajan, M. Mukund (Eds.), Formal Models, Languages and Applications. Series in Machine Perception and Artificial Intelligence, 66: 195-211.

Lalitha D (2015) Rectangular Array Languages Generated by a Petri Net, Computational Vision and Robotics. Advances in Intelligent Systems and Computing, 332: 17-27.

Lalitha D, Rangarajan K, Thomas DG (2012) Rectangular Arrays and Petri Nets, Combinatorial Image Analysis. LNCS, 7655: 166-180.

Linqiang Pan K, Song B, Isawasan P, Subramanian KG (2016) Array P Systems with Parallel Rewriting and Tables of Context-Free Rules. Journal of Computational and Theoretical Nanoscience, 13: 3636–3642.

Mitra I, Roy Biswas G, Biswas Majee S (2014) Effect of Filler Hydrophilicity on Superdisintegrant Performance and Release Kinetics From Solid Dispersion Tablets of A Model BCS Class II Drug. International Journal of Pharmacy Research & Technology, 4: 28-33.

Nagoba, B. S., Mantri, S., Adgaonkar, B (2017) Topical insulin and wound healing. European Journal of General Medicine, 14(2), 57-57. doi: 10.29333/ejgm/81885

Păun Gh (2002) Membrane Computing: An Introduction, Springer-Verlag, Berlin, Heidelbrg.

Subramanian KG, Hemalatha S, Sri Hari Nagore C, Margernstern M (2007) On the power of P systems with parallel rewriting and conditional communication. Romanian Journal of Information Science and Technology, 10(2): 137-144.

Subramanian KG, Isawasan P, Venkat I, Pan L (2014) Parallel Array Rewriting P systems. Romanian Journal of Information Science and Technology, 17(1): 103-116.