Available online at www.sciencedirect.com

ScienceDirect

Progress in Natural Science 18 (2008) 617–622

Progress in Natural Science

www.elsevier.com/locate/pnsc

Special report

P systems generating iso-picture languages

Subbaiah Annadurai [a,*], Thiyagarajan Kalyani [a], Vincent Rajkumar Dare [b], Durairaj Gnanaraj Thomas [b]

[a] *Department of Mathematics, St. Joseph's College of Engineering, Chennai 600119, India*

[b] *Department of Mathematics, Madras Christian College, Tambaram, Chennai 600059, India*

Received 21 December 2007; received in revised form 8 January 2008; accepted 10 January 2008

Abstract

P systems generating rectangular arrays and hexagonal arrays have been studied in the literature, bringing together the two areas of theoretical computer science, namely membrane computing and picture languages. Recently, a new class of picture languages called the class of iso-picture languages generating interesting picture languages has been introduced. In this paper, we develop a class of tissue-like *P* systems with active membranes as a generative device for iso-picture languages.

Keywords: *P* system; Iso-picture; Languages

1. Introduction

The study of syntactic methods of describing pictures considered as connected, digitized finite arrays in a two-dimensional plane have been of great interest. Picture languages generated by array grammars or recognized by array automata have been advocated since the 1970s for problems arising in the frame work of pattern recognition and image processing [1]. Motivated by these studies, we have introduced a new class of picture languages called iso-picture languages. The notions of local and recognizable iso-picture languages have been introduced in [2], inspired by the corresponding study in rectangular picture languages.

Iso-arrays are made up of isosceles right-angled triangles and an iso-picture is a picture formed by catenating iso-arrays of the same size. Iso-picture languages include more picture languages-like hexagonal picture languages, rectangular picture languages, and languages of rhombuses and triangles [3].

Membrane computing deals with distributed computing models inspired from the structure and the functioning of the living cell [4]. Very briefly, in the compartments defined by a hierarchical arrangement of membranes, one processes multisets of objects by evolution rules associated with the membranes. One of the branches of membrane computing is concerned with objects described by strings, and then one considers usual sets of strings instead of multisets of objects. These strings are processed by rewriting or other string handling operations. Recently, Ceterchi et al. [5] proposed a variant of tissue-like *P* systems in order to generate two-dimensional picture languages on rectangular grids. The interest of the study [5] is that it uses a novel technique of allowing the membranes themselves to hold the pixels of the pictures instead of the membranes just acting as regions for computation. This approach is extended to hexagonal picture languages in [6].

In this paper, the approach of [5] is extended to iso-picture languages, and analogous *P* systems for their generation are considered by taking into account that each element of an iso-picture (rhombus) has eight neighbors, two on its own row and three on the rows above and below

[*] Corresponding author. Tel.: +91 44 24501060; fax: +91 44 24500861. *E-mail address:* annadurai_70@yahoo.co.in (S. Annadurai).

1002-0071/$ - see front matter © 2008 National Natural Science Foundation of China and Chinese Academy of Sciences. Published by Elsevier Limited and Science in China Press. All rights reserved.
doi:10.1016/j.pnsc.2008.01.004

it. This enables to handle local and recognizable iso-picture languages introduced in [2].

2. Preliminaries

In this section, we recall the notions of iso-pictures, iso-picture languages and iso-triangular tiling systems proposed in [2].

Let $\Sigma = \{ {}^{a_1}A^{a_3}_{a_2},\ {}^{b_2}_{b_3}B_{b_1},\ {}^{c_3}_{c_1}C\,c_2,\ d_2\,D^{d_1}_{d_3} \}$ be a finite set of labeled isosceles right-angled triangular tiles of dimensions $\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}$ and 1 unit, obtained by intersecting a unit square by its diagonals.

Definition 1. An iso-array of size m $(m \geqslant 1)$ is an isosceles right-angled triangular arrangement of elements of Σ, whose equal sides are denoted as S_1 and S_3, and the unequal side as S_2. It consists of m tiles along the side S_2 and it contains m^2 gluable elements of Σ. Iso-arrays can be classified as U-iso-array, D-iso-array, R-iso-array and L-iso-array, if tiles A, B, D and C are used in side S_2, respectively.

Definition 2. Let Σ be a finite alphabet of iso-triangular tiles. An iso-picture of size $(n, m), n, m \geqslant 1$ over Σ is a picture formed by catenating n-iso-arrays of size m. The number of tiles in any iso-picture of size (n, m) is nm^2.

An element of an iso-picture p of size (n, m) is represented as $p(i, j, k)$, where i is the ith iso-array of the picture and j is the jth row of the ith iso-array and k is the kth element of jth row of the ith iso-array, where $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$ and $k = 1, 2, \ldots, 2j-1$. The set of all iso-pictures over the alphabet Σ is denoted by Σ_I^{**}. An iso-picture language L over Σ is a subset of Σ_I^{**}.

Definition 3. Let p be an iso-picture of size (n, m). We denote by $B_{n',m'}(p)$ the set of all sub iso-pictures of p of size (n', m'), where $n' \leqslant n, m' \leqslant m$. $\hat{p}$ is an iso-picture obtained by surrounding p with special boundary symbols #A, #B, #C, #D $\notin \Sigma$.

Definition 4. An iso-picture language $L \subseteq \Sigma_I^{**}$ is called local if there exists a finite set θ of iso-arrays of size 2 over $\Sigma \cup \{$#A, #B, #C, #D$\}$ such that $L = \{p \in \Sigma_I^{**} / B_{1,2}(\hat{p}) \subseteq \theta\}$ and is denoted by $L(\theta)$.

The family of local iso-picture languages will be denoted by *ILOC*.

Definition 5. Let $p \in \Sigma_I^{**}$ be an iso-picture. Let Σ and Γ be two finite alphabets and $\pi : \Gamma \to \Sigma$ be a mapping which we call, a projection. The projection by mapping π of the picture p is the picture $p' \in \Sigma_I^{**}$ such that $p'(i, j, k) = \pi(p(i, j, k))$ for all $1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m$, $1 \leqslant k \leqslant 2j-1$, where (n, m) is the size of the iso-picture p. In this case $p' = \pi(p)$.

Definition 6. Let $L \subset \Gamma_I^{**}$ be an iso-picture language. The projection by mapping π of L is the language $L' = \{p'/p' = \pi(p),\ \forall p \in L\} \subseteq \Sigma_I^{**}$. We denote by $\pi(L)$ the projection by mapping π of an iso-picture language L.

Definition 7. Let Σ be a finite alphabet. An iso-picture language $L \subseteq \Sigma_I^{**}$ is called recognizable if there exists a local iso-picture language L' over an alphabet Γ and a mapping $\pi : \Gamma \to \Sigma$ such that $L = \pi(L')$.

The family of all recognizable iso-picture languages will be denoted by *IREC*.

Example 1. Let L' be the iso-picture language of rhombuses, where the diagonals are represented by the tiles A2 and B2, and the tiles in the remaining positions are represented by tiles A1 and B1, a member of which is shown in Fig. 1.

This is a local picture language.

3. Tiling iso-picture languages

We define a two-dimensional iso-triangular tiling system as defined in [5]. We consider $\mathcal{P} = \{P_1, P_2, \ldots, P_k\}$ a finite collection of finite connected subsets of Z^2, called iso-prototiles. The iso-prototiles can be normalized such that the lexicographically least point is the origin $(0, 0) \in Z^2$.

A translate of an iso-prototile $P \in \mathcal{P}$ by a $t \in Z^2$ is the subset $t + P$ of Z^2 (where + is the addition in Z^2) and is called a triangular tile.

An iso-triangular tiling of the integer plane Z^2 by the iso-prototiles $\mathcal{P}$ is an expression of Z^2 as a disjoint union of triangular tiles, $Z^2 = \cup(t_j + P_{k_j})$, where $(t_i + P_{k_i}) \cap (t_j + P_{k_j}) = \phi$ if $i \neq j$. We equivalently say that the set of iso-prototiles $\mathcal{P}$ tiles the integer plane.

Consider $A = \{1, 2, \ldots, k\}$ the finite set of labels of iso-prototiles of $\mathcal{P}$ and think of it as a finite alphabet of shapes.

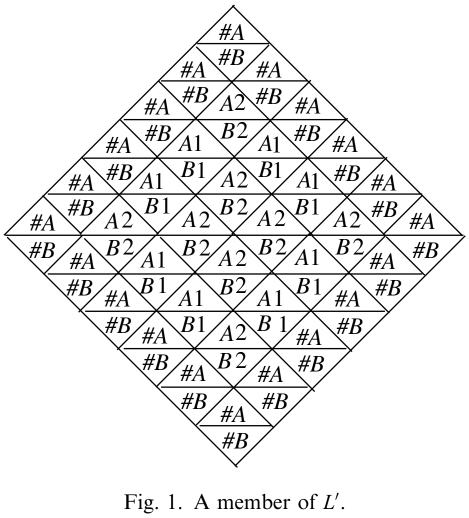

#A
#B
#A #A
#A #B A2 #B #A
#A #B A1 B2 A1 #B #A
#A #B A1 B1 A2 B1 A1 #B #A
#A #B A2 B1 A2 B2 A2 B1 A2 #B #A
#B B2 A1 B2 A2 B2 A1 B2 #A #B
#B #A B1 A1 B2 A1 B1 #A #B
#B #A B1 A2 B 1 #A #B
#B #A B2 #A #B
#B #A #B
#B

Fig. 1. A member of L'.

Consider A^{Z^2} the set of functions defined on Z^2 and with values in A. A tiling of the plane $Z^2 = \cup(t_j + P_{k_j})$ will be associated with an element $x \in A^{Z^2}$ in the following way: for every $v \in Z^2, x(v) = r$ iff the point v lies in a tile that is a translate of P_r, i.e., iff $v \in t_j + P_{k_j}$ and $k_j = r$. For $x \in A^{Z^2}$ and $v \in Z^2$, we denote $x(v)$ by x_v.

A two-dimensional shift is an application $\sigma : Z^2 \to Homeo(A^{Z^2})$ such that for any $v \in Z^2$, $\sigma_v : A^{Z^2} \to A^{Z^2}$ is the translation of the plane by the vector v, i.e., $(\sigma_v(x))_w = x_{w+v}$, for any $x \in A^{Z^2}$ and any $w \in Z^2$. A subset $X \subseteq A^{Z^2}$ is called σ-invariant iff $\sigma_v(x) \subseteq X$, for any $v \in Z^2$. Consider now the set $T(\mathcal{P})$ of all $x \in A^{Z^2}$ which corresponds to the iso-tiling of Z^2 by $\mathcal{P}$. This set is a σ-variant closed subset of A^{Z^2}, and thus $(T(\mathcal{P}), \sigma)$ is a subshift (A^{Z^2}, σ).

We extend the set of iso-prototiles $\mathcal{P}$ with one more iso-prototile $\mathcal{P}_{\#} = \{(0,0)\}$. We will think about the tiles obtained by translating this iso-prototile filled with $\#A$ or $\#B$. Then $\overline{\mathcal{P}} = \{P_1, P_2, \ldots, P_k, P_{\#}\}$ is the extended set of tiles, $\overline{A} = \{1, 2, \ldots, k, \#A, \#B\}$ is the extended alphabet of shapes.

Formally if $\overline{P} = \{P_1, P_2, \ldots, P_{\#}\}$ is a set of iso-prototiles, V is an alphabet and $\#A, \#B$ are special boundary symbols, denoting the blank, then for any $i = 1, 2, \ldots, k$, we can define the application $f_i : P_i \to V$, which associates a symbol from V with any pixel of P_i. For $P_{\#}$ we take $f_{\#} : P_{\#} \to \{\#A, \#B\}$. Then for any iso-tiling of the integer plane $Z^2 = \cup(t_j + P_{k_j})$, we can define an application $f : Z^2 \to V \cup \{\#A, \#B\}$ such that $f(t_j + w) = f_{k_j}(w)$, for any j and any $w \in P_{k_j}$. If the tiling is compatible with an iso-tiling, we can define an iso-picture $p = (P_{ijk})_{1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m, 1 \leqslant k \leqslant 2m-1}$ such that P_{ijk} represents exactly the value in V of the same pixel in the shift invariant iso-tiling. We denote by $\mathcal{L}(\mathcal{P}, V, \mathcal{F})$ the two-dimensional iso-picture language of elements of V_I^{**} coverable by tilings compatible with iso-tilings over a set of iso-prototiles $\mathcal{P} = \{P_1, P_2, \ldots, P_k\}$, an alphabet V and the applications

$$\mathcal{F} = \{f_1 : P_1 \to V, \ldots, f_k : P_k \to V\}.$$

Example 2. We consider a set of iso-prototiles $\mathcal{P} = \{P_1 = \{(0,0), (1,0)\}$, $P_2 = \{(1,0), (2,0)\}$, $P_3 = \{0,0\}$, $P_4 = \{(1,0)\}$. The alphabet $V = \{$△A, ▽B$\}$ and the applications $\mathcal{F} = \{f_1 : P_1 \to V, f_2 : P_2 \to V, f_3 : P_3 \to V, f_4 : P_4 \to V\}$, with $f_1(0,0) =$ △A, $f_1(1,0) =$ ▽B, $f_2(1,0) =$ ▽B, $f_2(2,0) =$ △A, $f_3(0,0) =$ △A, $f_4(1,0) =$ ▽B. The language $L_2 = L(\mathcal{P}, V, \mathcal{F})$ contains all the iso-pictures that can be covered with the iso-tiles represented in Fig. 2.

The iso-picture shown in Fig. 3 belongs to the language L_2.

Consider the local iso-picture language given in Example 1, the language of rhombuses. This local iso-picture language is generated by the set of iso-prototiles shown in Fig. 4.

Now we give the mapping as follows:

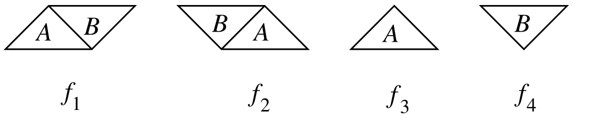


Fig. 2. Examples of iso-prototiles.

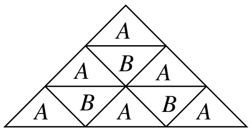
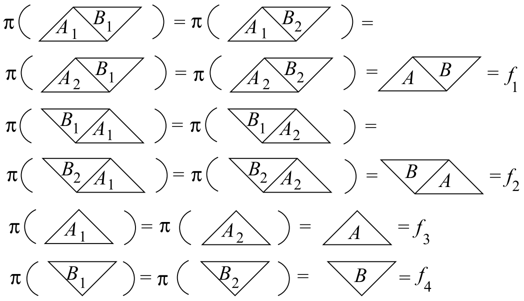


Fig. 3. Examples of an iso-picture that belongs to the language L_2.


π(A_1 B_1) = π(A_1 B_2) =
π(A_2 B_1) = π(A_2 B_2) = A B = f_1
π(B_1 A_1) = π(B_1 A_2) =
π(B_2 A_1) = π(B_2 A_2) = B A = f_2
π(A_1) = π(A_2) = A = f_3
π(B_1) = π(B_2) = B = f_4

Now it can be easily seen that the iso-picture language of rhombuses over the two-letter alphabet $\{$△A, ▽B$\}$ is the recognizable iso-picture language. Hence L_2 is a recognizable iso-picture language.

4. *P* systems for generating iso-picture languages

For generating two-dimensional pictures, tissue-like P systems [7] with active membranes were proposed in [5,6]. We apply this formalism of P systems to generate iso-pictures (for simplicity, let us take the iso-picture language of rhombuses, the same technique can be used to generate any iso-picture). The basic idea in the construction of the P system is on the lines of [5]. The difference lies mainly in the formation of rules.

A P system with active membranes is a construct $\pi = (O, H, \mu, w_1, \ldots, w_m, R)$, where O is the alphabet of objects, H is a finite set of labels for membranes, μ is a membrane structure consisting of m membranes, labeled with elements of H, $w_1, w_2, \ldots, w_m$ are strings over O representing the multiset of objects placed in the m regions of μ and R is a finite set of developmental rules. The rules of R are object evolution rules, communication rules, division rules and dissolving rules.

Let us consider a set of iso-prototiles $\{P_1, P_2, \ldots, P_k\}$, an alphabet V and the applications $f_1 : P_1 \to V, \ldots, f_k : P_k \to V$. We denote by Q the following set of headed iso-prototile labels

$$Q = \{(\alpha, \beta, \gamma) / (\beta, \gamma) \in P_\alpha\}$$

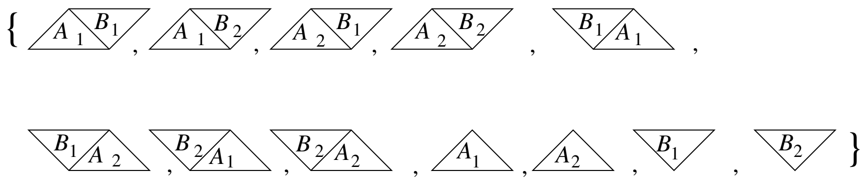


Fig. 4. Iso-prototiles for the local iso-picture languages given in Example 1.

We denote by $Q_0 \subseteq Q$ a distinguished subset of headed iso-prototile labels.

An iso-picture is generated by evolving an initial P system into a stable P system. An iso-picture is the result of the entire configuration of the final stable P system with each membrane corresponding to a pixel of an iso-picture. The communication graph of this P system is a rhombus grid in which each inner node is connected with four neighbors, each border node with three neighbors and each corner with two neighbors.

We distinguish 9 types of nodes in order to guarantee that this P system generates only iso-pictures (rhombuses). We denote by $T = \{ij/i, j \in \{0, 1, 2, 3, 4\}\}$ the set of labels corresponding to these nodes (Fig. 5). These 9 grid positions in the P systems can be represented by a set of 9 symbols $\{P_{02}, P_{11}, P_{13}, P_{20}, P_{22}, P_{24}, P_{31}, P_{33}, P_{42}\}$, such that at any moment, in any membrane in the system, at least one of the symbols is present. Every membrane in the system, after its creation, checks for the type of its neighbors, by communicating them to a symbol, which represents a possible neighbor type.

For example, a node on the right-up margin (margin holding the symbols P_{02}, P_{13}, P_{24}) of the grid may have as a left-down neighbor either an inner node or a node on the left-down margin (margin holding the symbols P_{20}, P_{31}, P_{42}) of the grid. Then a membrane containing the symbol P_{13} sends non-deterministically to its left-down neighbor either a symbol P_{22} or a symbol P_{31}. If two different symbols P_t and P_s are present at the same time in a membrane, then a killer (a special symbol k) is produced in that membrane and during the next evolution step the killer dissolves the membrane. Only those P systems having a rhombus grid as the communication graph will be stable.

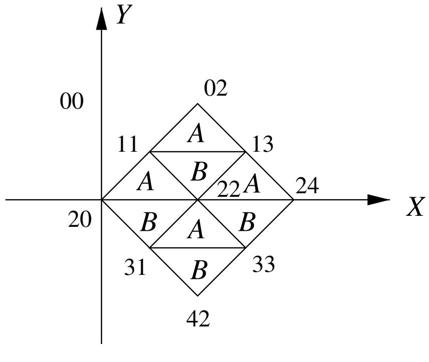


Fig. 5. Rectangular grid and the marked positions of a rhombus.

In this way, we will check by local tests for the whole integrity of the grid.

The generation of any iso-picture will start from its upper corner. Thus any stable P system will evolve from an initial P system, containing only one membrane and two symbols, the marker P_{02} and the output start symbol (a special symbol s).

The output start symbol evolves, in any membrane m in which it is present, either in $\#A$, $\#B$ if m is a membrane on the border of the iso-picture, or in an arbitrary symbol V if m is an inner membrane.

A membrane label from the set M is a multiset over a two-letter alphabet $\{r_d, l_d\}$. Considering that r_d and l_d indicate right-down and left-down, respectively, the label of the membrane will represent the path (s), which lead (s) from the upper corner of the grid (the λ membrane) to the given membrane. In this way, positions can be communicated with respect to the given membrane ($+r_d$ for right-down, $+l_d$ for left-down, $-r_d$ for left-up and $-l_d$ for right-up directions).

The rules are object evolution rules, dissolving rules, object evolution rules combined with membrane division and/or communication by symport rules and membrane division rules combined with communication by symport rules and dissolving rules.

Formally, an iso-tiling P system is a construct

$A = (O, V, M, (\mathrm{cont}_m)_{m \in M}, R)$ such that

- $O = V \cup \{\#A, \#B\} \cup \{P_t/t \in T\} \cup Q \cup \{s, c, k\}$ is the alphabet of symbol-objects. The symbols $(P_t)_{t \in T}$ indicate the type of the membrane with respect to the position into the rhombus grid. The subset Q represents the headed iso-prototile labels. The symbols s, c and k are the output start symbol, the checking start symbol and the killer, respectively.
- $V \subseteq O$ is the output alphabet.
- $M \subseteq (r_d + l_d)^*$. A membrane label is a multiset of support $\{r_d, l_d\}$ described by a string over r_d and l_d. If $m \in M$ and $x \in \{r_d, l_d\}$, then by $m - x$ we mean the multiset obtained from m by deleting one occurrence of x and by $m + x$ we mean the multiset obtained from m by adding one occurrence of x.
- The set of rules R is divided into three groups, as described below.

Creation rules: These rules are creating the grid of membranes, checking for the integrity of the grid and generating tiles on the grid.

1. $[P_{02}s]_m \rightarrow P_{02}\#B[P_{lk}s]_{m+l_d}[P_{nt}s]_{m+r_d}$ with $lk = \{11, 20\}$, $nt = \{13, 24\}$. The output start symbol s, in the presence of the marker P_{00}, evolves to $\#B$. Because of the presence of the marker P_{02}, m should be the membrane λ (the upper corner of the grid), which has two neighbors, right-down and to the left-down. The left-down neighbor should be a membrane either on the left-up margin or in the down corner of the left-up margin. Thus m divides to produce the membrane $m + l_d$ with the content $P_{11}s$ or $P_{20}s$. A similar effect is taken for the membrane $m + r_d$.
2. $[P_{11}s]_m \rightarrow P_{11}\#B[P_{lk}]_{m-l_d}[P_{uv}s]_{m+r_d}[P_{nt}s]_{m+l_d}$ with $lk = \{11, 02\}$, $nt = \{11, 20\}$, $uv = \{22, 33\}$. The output start symbol s, in the presence of the marker P_{11}, evolves to $\#B$. Because of the presence of the marker P_{11}, m should be a membrane on the left-up margin of the grid which has three neighbors right-up, right-down and left-down. The right-up neighbor already exists, thus m will send either P_{02} or P_{11} to the membrane $m - l_d$ in order to check its type. For the right-down neighbor, m concurs for its creation with another membrane in the system, namely $m - l_d + r_d$. If m wins, then it divides to produce $(m - l_d) + rd$ and declares its type by writing either $P_{22}s$ or $P_{33}s$ in it. If not this part of the rule is treated as a symport rule and m will send to the membrane $m + r_d$ either $P_{22}s$ or $P_{33}s$ in order to check its type. Finally m divides to produce the membrane $m - r_u$ with the content $P_{11}s$ or $P_{20}s$.
3. $[P_{20}s]_m \rightarrow P_{20}\#A[P_{lk}]_{m-l_d}[P_{nt}s]_{m+r_d}$ with $lk = \{11, 02\}$, $nt = \{31, 42\}$. Analogously with the rules 1 or 2.
4. $[P_{13}s]_m \rightarrow P_{13}\#B[P_{lk}]_{m-r_d}[P_{nt}s]_{m+r_d}[P_{uv}s]_{m+l_d}$ with $lk = \{13, 02\}$, $nt = \{13, 24\}$, $uv = \{22, 31\}$. Analogously with the rules 1 or 2.
5. $[P_{22}s^2]_m \rightarrow P_{22}f_\alpha(\beta, \gamma)$ $[P_{lk}]_{m-r_d}$ $[P_{nt}]_{m-l_d}$ $[P_{uv}y]_{m+r_d}$ $[P_{ab}z]_{m+l_d}$ with $lk = \{22, 11\}$, $nt = \{22, 13\}$, $uv = \{22, 33\}$, $ab = \{22, 31\}$. Analogously with the rules 1 or 2, with the only notable difference that s^2 evolves to $f_\alpha(\beta, \gamma)$ for a headed iso-prototile label $(\alpha, \beta, \gamma) \in Q_0$. In the new membranes $m + r_d$ and $m + l_d$, apart from the position markers, new objects are written depending on the headed iso-prototile label (α, β, γ). If $(\beta + 1, \gamma + 1) \in P_\alpha$, then $y = (\alpha, \beta + 1, \gamma + 1)$ else $y = s$. If $(\beta + 1, \gamma - 1) \in P_\alpha$, then $z = (\alpha, \beta + 1, \gamma - 1)$ else $z = s$.
6. $[P_{22}s(\alpha, \beta, \gamma)]_m \rightarrow P_{22}f_\alpha(\beta, \gamma)$ $[P_{lk}]_{m-r_d}[P_{nt}]_{m-l_d}$ $[P_{uv}y]_{m+r_d}$ $[P_{ab}z]_{m+l_d}$with $lk = \{22, 11\}$, $nt = \{22, 13\}$, $uv = \{22, 33\}$, $ab = \{22, 31\}$. Analogously with the rules 1 or 2, with the only notable difference that $s(\alpha, \beta, \gamma)$ evolves to $f_\alpha(\beta, \gamma)$ for a headed iso-prototile label $(\alpha, \beta, \gamma) \in Q_0$. In the new membranes $m + r_d$ and $m + l_d$, apart from the position markers, new objects are written depending on the headed iso-prototile label (α, β, γ). If $(\beta + 1, \gamma + 1) \in P_\alpha$, then $y = (\alpha, \beta + 1, \gamma + 1)$ else $y = s$. If $(\beta + 1, \gamma - 1) \in P_\alpha$, then $z = (\alpha, \beta + 1, \gamma - 1)$ else $z = s$.
7. $[P_{22}(\alpha, \beta, \gamma)^2]_m \rightarrow P_{22}f_\alpha(\beta, \gamma)$ $[P_{lk}]_{m-r_d}$ $[P_{nt}]_{m-l_d}$ $[P_{uv}y]_{m+r_d}$ $[P_{ab}z]_{m+l_d}$ with $lk = \{22, 11\}$, $nt = \{22, 13\}$, $uv = \{22, 33\}$, $ab = \{22, 31\}$. Analogously with the rules 1 or 2, with the only notable difference that $(\alpha, \beta, \gamma)^2$ evolves to $f_\alpha(\beta, \gamma)$ for a headed iso-prototile label $(\alpha, \beta, \gamma) \in Q_0$. In the new membranes $m + r_d$ and $m + l_d$, apart from the position markers, new objects are written depending on the headed iso-prototile label (α, β, γ). If $(\beta + 1, \gamma + 1) \in P_\alpha$, then $y = (\alpha, \beta + 1, \gamma + 1)$ else $y = s$. If $(\beta + 1, \gamma - 1) \in P_\alpha$, then $z = (\alpha, \beta + 1, \gamma - 1)$ else $z = s$.
8. $[P_{31}s]_m \rightarrow P_{31}\#Ac[P_{lk}]_{m-r_d}[P_{nt}]_{m-l_d}[P_{uv}s]_{m+r_d}$ with $lk = \{20\}$, $nt = \{22, 13\}$, $uv = \{31, 42\}$. Analogously with the rules 1 or 2.
9. $[P_{24}s]_m \rightarrow P_{24}\#Ac[P_{lk}]_{m-r_d}[P_{nt}s]_{m+l_d}$ with $lk = \{02, 13\}$, $nt = \{33, 42\}$. Analogously with the rules 1 or 2.
10. $[P_{33}s]_m \rightarrow P_{33}\#Ac[P_{lk}]_{m-r_d}[P_{nt}]_{m-l_d}[P_{uv}s]_{m+l_d}$ with $lk = \{33, 22\}$, $nt = \{24\}$, $uv = \{33, 42\}$. Analogously with the rules 1 or 2.
11. $[P_{42}s]_m \rightarrow P_{42}\#Ac[P_{lk}]_{m-r_d}[P_{nt}s]_{m-l_d}$ with $lk = \{20, 31\}$, $nt = \{33, 24\}$. Analogously with the rules 1 or 2. Contamination rules: These rules are contaminating the P system with the killer that dissolves the membranes (by destruction rules).
12. $[P_{ij}P_{lk}]_m \rightarrow k$ with $i, j, l, k \in \{0, 1, 2, 3, 4\}$ and $(i, j) \neq (l, k)$. Two different markers in the same membrane produce a killer.
13. $[(\alpha, \beta, \gamma)(\alpha', \beta', \gamma')]_m \rightarrow k$ with (α, β, γ), $(\alpha', \beta', \gamma') \in Q$, $(\alpha, \beta, \gamma) \neq (\alpha', \beta', \gamma')$. Two different headed iso-prototile labels in a boundary membrane produce a killer.
14. $[P_{ij}(\alpha, \beta, \gamma)]_m \rightarrow k$ with $ij \neq 22$ and $(\alpha, \beta, \gamma) \in Q$. A headed iso-prototile label in a boundary membrane produces a killer. Destruction rules: By these rules, the killer spreads from a contaminated membrane all over the P system, while dissolving the membranes in which it appears.
15. $[P_{02}k]_m \rightarrow [k]_{m+r_d}[k]_{m+l_d}\delta$. The killer in the presence of the marker P_{02} is sent right-down and to left-down and dissolves the membrane m.
16. $[P_{11}k]_m \rightarrow [k]_{m-l_d}[k]_{m+r_d}[k]_{m+l_d}\delta$. The killer in the presence of the marker P_{11} is sent right-down, left-down and right-up, and dissolves the membrane m.
17. $[P_{20}k]_m \rightarrow [k]_{m-l_d}[k]_{m+r_d}\delta$. Analogously with the rules 15 or 16.
18. $[P_{13}k]_m \rightarrow [k]_{m-r_d}[k]_{m+r_d}[k]_{m+l_d}\delta$. Analogously with the rules 15 or 16.
19. $[P_{22}k]_m \rightarrow [k]_{m-l_d}[k]_{m-r_d}[k]_{m+r_d}[k]_{m+l_d}\delta$. Analogously with the rules 15 or 16.
20. $[P_{31}k]_m \rightarrow [k]_{m-r_d}[k]_{m-l_d}[k]_{m+r_d}\delta$. Analogously with the rules 15 or 16.
21. $[P_{24}k]_m \rightarrow [k]_{m-r_d}[k]_{m+l_d}\delta$. Analogously with the rules 15 or 16.
22. $[P_{33}k]_m \rightarrow [k]_{m-r_d}[k]_{m-l_d}[k]_{m+l_d}\delta$. Analogously with the rules 15 or 16.
23. $[P_{42}k]_m \rightarrow [k]_{m-r_d}[k]_{m-l_d}\delta$. Analogously with the rules 15 or 16.

An iso-tiling P system $A = (O, V, M, (\text{cont}_m)_{m \in N}, R)$ is called initial over V iff $M = \{\lambda\}$ and $\text{cont}_\lambda = \{P_{02}s\}$. With any stable alive iso-tiling P system $A = (O, V, M, (\text{cont}_m)_{m \in M}, R)$ we may associate an iso-picture over the alphabet V in the following way.

1. First, we define two natural numbers, s and t by:

$s = \max\{i | \exists j \text{ such that } r_d^i l_d^j \in M\} - 1,$

$t = \max\{j | \exists i \text{ such that } r_d^i l_d^j \in M\} - 1.$

2. If s and t are greater than 1, then we consider an iso-picture

$(P_{ijk})_{1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m, 1 \leqslant k \leqslant 2m-1}$ with $P_{ijk} = h$ iff $h \in \text{cont}_{r_d^i l_d^j} \cap V$; otherwise, we consider the empty picture. We define an iso-picture language generated by an initial grid-like P system with active membrane A, as the set of iso-pictures associated with all grid-like P systems with active membranes from the stable universe of A. Thus we obtain the following result.

Theorem 1. *An initial grid-like P system with active membranes, over a given output alphabet V, without any checking rules generates the language of all iso-pictures over V.*

In particular, the generation of local and recognizable iso-picture language introduced in [2] *can be done with grid-like P systems with active membranes. The details are omitted as it can be done analogously to the rectangular case* [5].

We end this section, considering the iso-picture language $L_2 = L(\mathcal{P}, V, \mathcal{F})$ defined in Example 2. Using the above notations we can define an initial iso-tiling P system A with the set of headed iso-prototiles labels

$Q = \{(1,0,0), (1,1,0), (2,1,0), (2,2,0), (3,0,0), (4,1,0)\}$

and the subset of starting headed iso-prototiles labels

$Q_0 = \{(1,0,0), (2,1,0), (3,0,0), (4,1,0)\}.$

Then the language of iso-pictures associated with all the iso-tiling P systems from the stable universe of A is exactly L_2.

5. Conclusions

The generation of iso-pictures by P systems based on the techniques of [5] is considered here. Although the extension is done for an iso-picture language of rhombuses, the study can be extended to generate any general iso-picture language.

References

[1] Giammarresi D, Restivo A. Recognizable picture languages. Intl J Pattern Recognit Artif Intell Parallel Image Process (Special Issue) 1992:31–46.

[2] Kalyani T, Dare VR, Thomas DG. Local and recognizable iso-picture languages. LNCS 2004;3316:738–43.

[3] Kalyani T, Dare VR, Thomas DG. Iso-array acceptors and learning. LNAI 2006;4201:327–39.

[4] Păun Gh. Computing with membranes. J Comput Syst Sci 2000;61(1):108–43.

[5] Ceterchi R, Gramatovici R, Jonoska N, et al. Tissue-like *P* systems for picture generation. Fundam Informat 2003;56:311–28.

[6] Dersanambika KS, Krithivasan K, Subramanian KG. *P* systems generating hexagonal picture languages. Workshop Membr Comput 2003:168–80.

[7] Păun A, Păun Gh, Rozenberg G. Computing by communication in networks of membranes. Intl J Found Comput Sci 2002;13(6):779–98.