ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **15**, Number 3, 2012, 246–265

Generative models for pictures tiled by triangles

K. G. SUBRAMANIAN[1], P. WIEDERHOLD[2], Ibrahim VENKAT[1], Rosni ABDULLAH[1]

[1] School of Computer Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia

E-mail: `kgsmani1948@yahoo.com, ibrahim@cs.usm.my, rosni@cs.usm.my`

[2] Department of Automatic Control, Centro de Investigación y de Estudios Avanzados (CINVESTAV-IPN), Av. I.P.N. 2508, Col. San Pedro Zacatenco, Mexico 07000 D.F., Mexico

E-mail: `pwiederhold@gmail.com`

Abstract. The problem of generation of triangle-tiled pictures that are two-dimensional figures tiled by isosceles right triangles of the same size, is investigated. Grammars called context-free triangle-tiled picture grammars (CF-ttPG) that generate such figures making use of context-free type rules, are proposed. The special case that all rules of such a grammar are regular, is studied. In the framework of membrane computing which is a biologically inspired computability model, generation of triangle-tiled pictures by rules of a CF-ttPG is considered. The picture language of rectangular hollow frames which cannot be generated by any CF-ttPG, can be generated by a specific type of P system with two membranes.

Key-words: picture generation by grammars, context-free grammar, picture grammar, triangle-tiled picture, P system, figure decomposed by triangles, rectangular hollow frame.

1. Introduction

In studies on image analysis, syntactic models of picture generation have occupied an important place. Several types of picture or array grammars [9,20,21,25] generating picture arrays in the two-dimensional plane have been introduced and investigated by researchers with different needs of application. These models generate picture arrays on a rectangular or other grids and are based on the techniques of formal string language theory [9]. Such a grammar not only generates certain patterns composed of array elements (such as pixels, or tiles of a predefined plane tiling), but additionally, each array element of the pattern has a label given by a symbol of a finite alphabet.

On the other hand, tiling the plane with a given finite set of tiles has always been a problem of interest. Grünbaum and Shephard [11] give fascinating examples of decorative tilings. Theoretical investigations of several questions related to tiling including generative grammars for figures tiled by unit squares are available (see, for example, [3–5, 10, 15–17]).

We consider here plane figures tiled by isosceles right triangles of the same size that are obtained by cutting a square by its diagonals. Another interpretation is that these plane figures are generated by gluing together congruent copies of an isosceles right angled triangular tile T, where these copies (including T itself) appear only in four distinct orientations corresponding to rotations of T by multiples of a right angle. Moreover, the tiles are labeled by symbols from a finite alphabet.

We refer to the sets of tiles forming these plane figures as triangle-tiled pictures, or in abbreviated form as tt-pictures. In this paper we investigate the problem of their generation by rules of grammars which are called as context-free and regular triangle-tiled picture grammars. These grammars are analogs of the well-known context-free and regular array grammars ([20, 21, 25]) which act on a pre-fixed tiling of the plane, where the tiles, for example, are squares or hexagons. Nevertheless, tt-pictures considered here have a richer structure than sets of tiles belonging to a pre-fixed (for example, triangular) tiling of the plane.

A new theoretical computability model introduced by Păun [18] which is called a P system and which is based on biological considerations, is known to be a versatile framework for studying many kinds of computational problems. In the basic model of a P system, multi-sets of objects are processed by evolution rules in regions defined by a hierarchical arrangement of membranes and communication can take place among the regions, sending objects from one to another. As an application of the P system model, Ceterchi et al [6] proposed an array-rewriting P system for picture generation by relating array-rewriting grammars and P systems. This study subsequently has given rise to a number of different kinds of array P systems (see, for example, [22] and references therein).

Here we study a triangle-tiled picture P system and examine its generative power. Also, we examine the feature of endowing a triangle-tiled picture P system with the t-mode of communication considered for array P systems in [23] and point out the power of the resulting system in the description of certain triangle-tiled pictures.

2. Basic Definitions

The generation of tt-pictures by a grammar is based in this paper on the following idea: We start with a very simple tt-picture (for example, a unique tile). Then an iterative growing process is performed on the tt-picture. In each iteration step only ONE rule of the grammar is applied, and this rule application consists in gluing one or several new admissible tiles to a tile which already belongs to the tt-picture. In the following we define our concepts of tile and of tt-picture, and we specify the possibilities of gluing tiles together. Our terminology and notions are mainly based on [12, 13, 19] but have been formalized and extended to our purposes.

2.1. Triangle-tiles

A compact connected subset of the Euclidean plane, named a ***figure***, is considered to be tiled by isosceles right triangles of four different types corresponding to the result of cutting a square by its diagonals, see Figure 1(a). Each of these triangles is called a ***triangle-tile*** or simply a ***tile***. Moreover, we suppose that each triangle-tile is labeled by a symbol taken from a finite alphabet. As an example of labeling, if the symbols a, b, c, d belong to that alphabet, Figure 1(b) shows the four types of triangle-tiles, each of them labeled by a symbol. As indicated in Figure 1(b), we name these types of tiles ***n-tile*** (north), ***s-tile*** (south), ***e-tile*** (east) and ***w-tile*** (west).

We refer to the catheti of an n-tile or an s-tile as the *left side* and the *right side* whereas the catheti of an e-tile or a w-tile are called the *top and the bottom sides*. Any cathetus of a triangle-tile or its hypotenuse is named a *side* of this triangle.

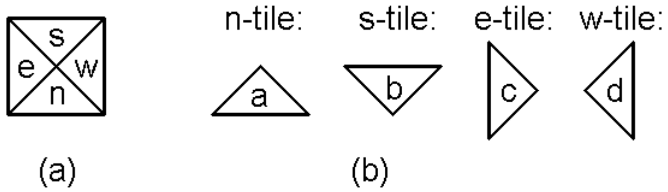

Fig. 1. The four types of labeled isosceles triangle-tiles.

2.2. Gluing of triangle-tiles

Gluing two triangle-tiles means to unify them along a whole side so that their interiors do not intersect and their union forms a new figure which is tiled by these two tiles. Let T be a (labeled) n-tile which in Figure 2 is represented as labeled by the symbol a. The other three types of (labeled) tiles can be glued with T in five ways as shown in Figure 2, where the labeling by the symbols b, c, d is only an example:

- There is a unique possibility to glue another tile V to T at the hypotenuse. Then V has to be an s-tile, see Figure 2(a).
- There are two possibilities to glue another tile V to T at the left side of T. As shown in parts (b) and (d) of Figure 2, V could be an s-tile or an e-tile. It is clear that we can apply only one of these two possibilities at the same time to

glue a tile V to T in order to generate some tt-picture which corresponds to a plane figure.

- Analogously, there are two possibilities to glue another tile V to T at the right side of T. As shown in parts (c) and (e) of Figure 2, V could be an s-tile or a w-tile. It is clear again that only one of these two possibilities should be permitted to be applied at the same time in some growing process which aims to generate a tt-picture.

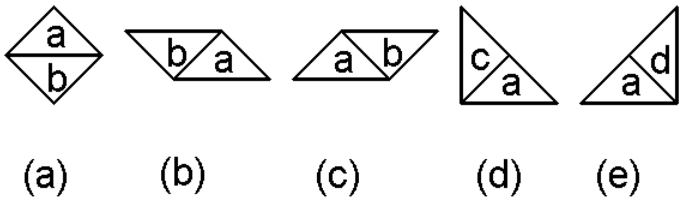

Fig. 2. Tiles glued to an n-tile.

Gluing of other pairs of the four types of (labeled) triangle-tiles can be similarly defined. There are always five possibilities of gluing a new tile V to some fixed tile T. Nevertheless, in order to generate a tt-picture within an iterative growing process, from the two possibilities to glue a new tile on a cathetus, only one has to be selected. As a result, only three other tiles can be glued at the same time to T, one at each triangle side.

We emphasize that this ambiguity of having five gluing possibilities from which only three of these can be applied at the same time is not present when working on the pre-fixed triangular plane tiling where all triangle tiles are congruent equilateral triangles. For this latter plane tiling, of course any tile can be glued exactly with other three tiles belonging to the plane tiling. In this paper we do not work on this (Archimedean) triangular tiling. We will clarify below which is the underlying set where our (new) tiles come from.

Another problem is that within an iterative growing process which generates a tt-picture, in each step there is an actual tt-picture M. The set M can grow only by gluing new tiles V which do not yet belong to M. In the context of generation by a grammar, each such new tile V is not labeled by any symbol of the alphabet which is used to label the tiles of M. That means that V belongs to the "background", and this fact is represented by having a special "background label". In consequence, at most three other tiles can be glued at a time to T, maximally one at each triangle side. In particular, when T is surrounded by other three tiles from M, no new tile V can be glued to T.

2.3. Triangle-tiled pictures

Any two distinct triangle-tiles are named ***adjacent*** if there share a common side. For any set M of (labeled) triangle-tiles and tiles $A, B \in M$, a path from A to B is a sequence $\{A = T_1, T_2, \cdots, T_m = B\}$ of (labeled) triangle-tiles such that any two tiles T_i, T_{i+1}, for $1 \leq i \leq m-1$, are adjacent. The set M is called ***connected*** if for any two tiles A, B of M, there is a path from A to B all of whose elements (tiles) belong to M. In the following we define formally a tt-picture and a figure.

Definition 1. A *triangle-tiled picture (tt-picture)* is a finite connected set of labeled triangle-tiles of the four types n-tile, s-tile, e-tile and w-tile (described above), whose labels belong to a finite alphabet. The compact connected subset of the Euclidean plane formed by the (point set) union of the tiles belonging to a tt-picture, is named a *figure* or a pattern.

From the comments in the previous subsection it follows that within a tt-picture, each tile can be adjacent to at most three other tiles. Figure 3 shows in (a) and (b) two examples of tt-pictures (the labels of the tiles are not shown in that figure). The triangle-tile set shown in Figure 3(c) is not a tt-picture since it is not connected by our definition: for example, from the tile labeled by p there is no path in the Figure to the tile labeled by t. The set of all tt-pictures over a fixed alphabet Σ is denoted by Σ^{t+}.

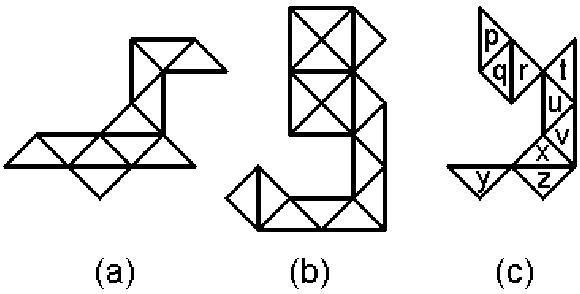

Fig. 3. (a) and (b): tt-pictures, (c): a disconnected set of triangle-tiles.

In this paper we are particularly interested in a special example of tt-picture which corresponds to the figure of a "thin hollow frame". Hollow frames constructed by quadratic or rectangular tiles have been useful as examples of pictures which cannot be generated by picture grammars having only context-free rules, see, for example, [26].

Definition 2. For any natural numbers $k, l \geq 1$, a *rectangular tt-hollow-frame* of size $k \times l$ is a tt-picture, whose triangle-tiles can be arranged into a (cyclic ordered) sequence as follows:
$\langle nb_1, sb_1, nb_2, sb_2, \cdots, sb_{k-1}, nb_k, wc_1, ec_1, wr_1, er_1, \cdots, er_{l-1}, wr_l, ec_2, wc_2,$
$st_1, nt_1, \cdots, nt_{k-1}, st_k, ec_3, wc_3, el_1, wl_1, \cdots, wl_{l-1}, el_l, wc_4, ec_4\rangle$.

Here for any subindex i, nb_i denotes an n-tile ("bottom", since it contributes to form the bottom base line of the rectangle), and the other notations mean the following: sb_i = s-tile ("bottom"), wr_i = w-tile ("right side"), er_i = e-tile ("right side"), st_i

= s-tile ("top side"), nt_i = n-tile ("top side"), el_i = e-tile ("left side"), wl_i = w-tile ("left side"), wc_i = w-tile ("corner"), ec_i = e-tile ("corner"). In this ordered cyclic sequence, any tile is adjacent with its immediate successor and with its immediate predecessor; but there are no other pairs of adjacent tiles. In the case $k = l$, the rectangular tt-hollow-frame is called a *square tt-hollow-frame* of side length k. Let ***tt-RHF*** and ***tt-SHF*** denote the set of all rectangular tt-hollow-frames and the set of all square tt-hollow-frames, respectively, where all tiles have the unique (terminal) label a.

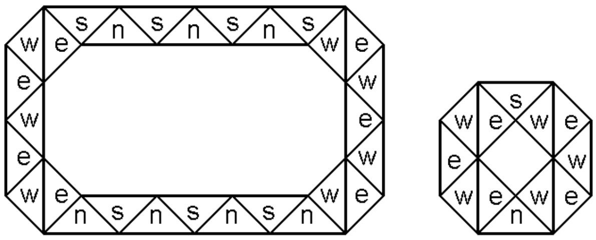

Fig. 4. A rectangular tt-hollow-frame (left), the smallest square tt-hollow-frame (right).

From the definition it is clear that within the cyclic sequence, the adjacency for each pair of tiles (nb_i, sb_i) or (wr_i, er_i) or (st_i, nt_i) or (wl_i, el_i) is given by a common cathetus. Also each corner tile wc_i or ec_i, is glued with a non-corner tile by such a side. Gluing by a common hypotenuse is only applied to any two corner tiles which are subsequent in the sequence. One member of tt-RHF is shown in the left hand side of Figure 4. It is a rectangular tt-hollow-frame of size 4×2, since four n-tiles form the bottom side (and four s-tiles form the top side) and two w-tiles form the right side (and two e-tiles form the left side). The right hand side of Figure 4 shows the smallest rectangular tt-hollow-frame, with $k = l = 1$, which is also the smallest square tt-hollow-frame. The tiles in Figure 4 are labeled by symbols which indicate the tile types, only for illustrating the previous definition of a rectangular tt-hollow-frame.

2.4. Tilings and tt-pictures

The tiles belonging to a tt-picture in general do not belong to only one plane tiling. We have already mentioned above that we do not work on the Archimedean triangular plane tiling. But it turns out that there is no triangular plane tiling which our tt-pictures belong to, even if we relax the condition on the triangle tiles to be equilateral. Let us understand a triangular plane tiling as a set of congruent (closed) triangles whose point set union covers the Euclidean plane and where any two triangles are disjoint or intersect in a whole common side. So, our triangle-tiles of course serve to form such tilings but these are two distinct ones:

Consider first the plane tiling $\mathcal{T}_1$ given by only n-tiles and s-tiles which are glued together using exclusively the gluing rules from Figures 2 (a),(b),(c). Analogously, let $\mathcal{T}_2$ be the plane tiling given by only e-tiles and w-tiles which are glued together using only the following three gluing rules: gluing an e-tile with a w-tile through their hypotenuse, gluing the top side of an e-tile with the bottom side of a w-tile, gluing the top side of a w-tile with the bottom side of an e-tile. Note that the tilings $\mathcal{T}_1$ and $\mathcal{T}_2$ are disjoint. Any tt-picture can be seen as a subset of the union $\mathcal{T}_1 \cup \mathcal{T}_2$, but in general a tt-picture does belong neither to $\mathcal{T}_1$ nor to $\mathcal{T}_2$. For example, the left hand side of Figure 5 contains the tt-picture example from Figure 3(a), whereas the center part presents the tiles belonging to the tiling $\mathcal{T}_1$, and the right hand side of Figure 5 shows the tiles which belong to $\mathcal{T}_2$.

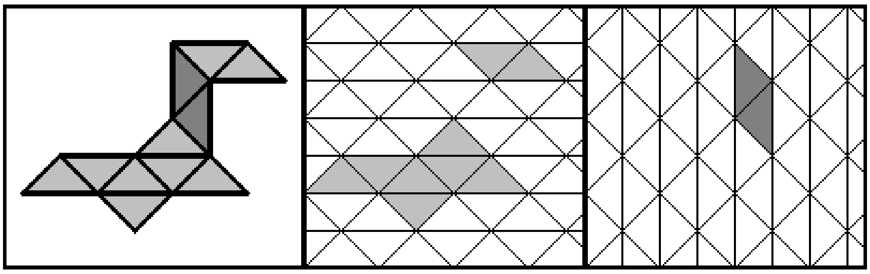

Fig. 5. A tt-picture belonging to the union of the two tilings $\mathcal{T}_1$ and $\mathcal{T}_2$.

Now, this paper aims to generate tt-pictures by grammar rules within an iterative growing process where in each iteration step only one rule is applied to an actual tt-picture M which is a subset of $\mathcal{T}_1 \cup \mathcal{T}_2$. The growing is achieved by gluing tiles of M with new tiles V which do not (yet) belong to M, that is, such tiles V belong to the "background". Now we are in position to define that for any given tt-picture $M \subset \mathcal{T}_1 \cup \mathcal{T}_2$, its ***background*** is the set of tiles given as $(\mathcal{T}_1 \cup \mathcal{T}_2) \setminus M$. All tiles belonging to the background will be supposed to be labeled with a special symbol named ***background symbol*** or ***blank symbol*** $\sharp$ which does not belong to the alphabet $\mathcal{A}$ which is used to label the tiles of a tt-picture of interest. Adding a background tile V to M by gluing it to some tile of M will mean to rewrite the blank symbol $\sharp$ by some symbol of the alphabet $\mathcal{A}$.

3. Context-free triangle-tiled picture grammar

The motivation for the generation of a tt-picture by a grammar is that a tt-picture models some figure of interest (in the Euclidean plane, or a corresponding object of interest within a two-dimensional digital image), which then is generated by a growing process starting with a simple tt-picture, for example, with a single labeled triangle-tile. In each growing step, the tiles belonging to an actual tt-picture are recognized since they are labeled by symbols belonging to some alphabet $\mathcal{A}$. The tt-picture grows up by gluing their tiles with additional new triangle-tiles from the background which can be recognized since they are labeled by the blank symbol $\sharp \notin \mathcal{A}$.

This same idea has been used in the development of the well-known picture array grammars which are applied to pre-fixed tilings of the plane: rectangular picture array grammars act on the quadratic plane tiling, whereas hexagonal array grammars are applied to the hexagonal plane tiling. The latter two plane tilings extend infinitely over the Euclidean plane, and any subset of tiles which already has been generated by the grammar, has a finite number of tiles whose union forms a figure being a bounded subset of the Euclidean plane. This fact is also true for our case of tt-pictures which belong to the union $\mathcal{T}_1 \cup \mathcal{T}_2$. In consequence, it can be supposed that for any given tt-picture M, there exist background tiles surrounding M completely.

We now define a triangle-tiled picture grammar, restricting our attention to the context-free and regular cases, although a general definition is possible as in an isometric array grammar for the rectangular grid as defined, for example, in [21, 25].

Definition 3. A *context-free triangle-tiled picture grammar (CF-ttPG)* is a grammar $G = (N, \Sigma, P, (t, S), \sharp)$ where N, Σ are disjoint finite sets of symbols (respectively called non-terminals and terminals), t is the start triangle-tile labeled by the start symbol S, $S \in N$, $\sharp$ is a special blank symbol or background symbol not in $N \cup \Sigma$. P is a finite set of rewriting rules of the form $\alpha \longrightarrow \beta$, where α, β are tt-pictures with their labels over the alphabet $N \cup \Sigma \cup \{\sharp\}$ satisfying the following conditions:

i) α, β are identical in their geometric shapes, which means that the figure corresponding to α is a translation of the figure corresponding to β;

ii) α contains exactly one tile labeled by an element of N whereas all other tiles (if any) in α have label $\sharp$;

iii) All tiles of β are labeled over $(N \cup \Sigma)$.

Observe that α and β are connected sets of tiles, due to the definition of a tt-picture; (t, S) is a special tt-picture. For any tt-picture having at least two elements, its connectivity implies that each tile is adjacent to at least one other tile; in other words, each tile has at least one of its adjacent tiles non-blank. The grammar rules are based on the gluing rules introduced earlier.

Remarks:

1. The kind of grammars considered in [1, 12] with the name "iso-array grammars" also use the type of triangle tiles considered here. But these grammars involve a feature of use of different kinds of blank symbols. A blank symbol is intended to be an indication of an unoccupied empty tile and so using different kinds of blank symbols amounts to the tile being labelled by a non-blank symbol and does not indicate that the tile is empty. In our formulation of tt-picture grammar we have only one blank symbol $\sharp$ as the label of an empty or background tile which indicates that a non-blank symbol can label this tile in a subsequent step of rewriting.

2. The definition of context-free tt-picture grammar was introduced in [24] but an error has crept into the definition which has been rectified here. The part *ii)* of the definition precisely indicates the context-freeness of the tt-picture grammar.

3. As we will see later, the fact that even for a context-free grammar, the left hand side of a rule is permitted to contain (an arbitrary but finite set of) background tiles (labeled by $\sharp$), causes certain context sensitivity. This is similar to the feature which was observed in [26] for array grammars acting on the rectangular plane grid.

4. The part *iii)* of the definition indicates that it is not possible to rewrite any label by the blank symbol. Hence, a tt-picture is never shortened in the growing process (no tiles are deleted from the tt-picture). Growing is stopped at a tile when this tile is rewritten by a terminal symbol.

A ***derivation*** in a context-free tt-picture grammar G is denoted by $\gamma \Longrightarrow \delta$, for $\gamma, \delta \in (N \cup \Sigma \cup \{\sharp\})^{t+}$ (that is, γ, δ are tt-pictures with the labels of the tiles in $N \cup \Sigma \cup \{\sharp\}$), and it means that there exists a rule $\alpha \longrightarrow \beta$ such that α is a sub-tt-picture of γ and δ is obtained from γ by replacing α by β. So, a derivation corresponds to one step in the iterative growing process of a tt-picture, and in this step only one grammar rule is applied.

The reflexive transitive closure of $\Longrightarrow$ is denoted by $\Longrightarrow^*$ as usual. The set $L(G)$ of all tt-pictures generated by the grammar G is the set of tt-pictures labeled over the set of terminal symbols Σ, derivable in one or more steps from the start tt-picture (t, S). $L(G)$ is called the ***(tt-picture) language*** generated by G.

Definition 4. A rule of a CF-ttPG is called *regular* if it is of one of the following two types:

Type 1: The left hand side of the rule consists of only one tile T labelled by a non-terminal symbol and the rule rewrites it by a terminal symbol.

Type 2: The left hand side of the rule has two tiles, one of which is a tile T labeled by a non-terminal symbol and another a background tile V (labeled by $\sharp$) which is adjacent to T. The rule rewrites the label of T by a terminal symbol and the label of V by a non-terminal symbol.

A regular rule of type 1 is a "stop-rule" since the application of this rule stops the tt-picture growing further. A regular rule of type 2 has the result that the tile V (originally from the background) is added to the tt-picture and can be further considered by other rules. Hence, applying such a rule makes the tt-picture grow up into the direction where the blank symbol is situated before the rule application. For each triangle-tile T there exist five other tiles which can be adjacent to T. Hence for each tile T there are five possible regular grammar rules of type 2. Figure 6 shows these rules and also the regular rule of tye 1 for an e-tile labeled by A, where A, B are non-terminals and a is a terminal symbol.

Definition 5. A context-free tt-picture grammar G is called a *regular triangle-tiled picture grammar (Reg-ttPG)* if all the rules are regular. The family of the languages of all tt-pictures which can be generated by any CF-ttPG is denoted by *CF-ttPL*. That is,

$$\text{CF-ttPL } = \{L(G) : \; G \;\text{ is a grammar of type CF-ttPG }\} \; .$$

Analogously, the family of the languages of all tt-pictures generated by any Reg-ttPG is denoted by *Reg-ttPL*.

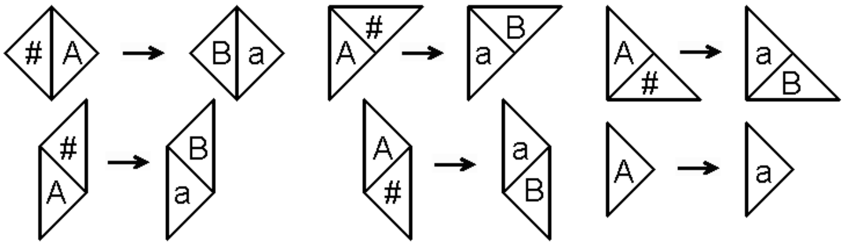

Fig. 6. The set of possible Reg-ttPG rules for an e-tile.

In the proof of Theorem 1 an example of a Reg-ttPG will be presented.

For picture array grammars (known before) acting on a pre-fixed tiling [25], for any tile T (for example, a square or a hexagon) of a picture already generated, if T is adjacent with a background tile V, then the picture can be grown up by writing a (non-blank) symbol on this uniquely defined tile V. Now, recalling the ambiguity of our gluing rules, for each triangle-tile T and each of its catheti, there are two tiles V_1, V_2 which could be glued with T by this side. Actually, these two rules are present in Figure 6 for each such side of an e-tile. This ambiguity does not cause contradictions since any CF-ttPG permits in each derivation the application of only one rule. Moreover, a particular grammar in general does not contain all possible rules.

However, we should mention that the applicability of a grammar rule has to be checked carefully in each derivation. The condition of having V as a background tile adjacent to a tile T (labeled by a non-terminal) which belongs to a tt-picture M is not sufficient for the applicability of a rule which pretends to add V to M. Even when V indicates an "empty place" to which M can be grown, it is possible that this "place" is not really empty: This happens when V is adjacent with T through a cathetus of T but there is another tile V_2 adjacent with T through the same cathetus and where V_2 is not empty but belongs to M. For example, observe the tt-picture M in the left part of Figure 7. The other parts of the figure show the tiles of M (all drawn in grey) belonging to the tilings $\mathcal{T}_1$ and $\mathcal{T}_2$. Certainly the tiles labeled by c, z, v, y and x do not belong to M, they are background tiles. Suppose that a grammar rule of a CF-ttPG consists in rewriting an n-tile T (labeled by a non-terminal) by a terminal and adding a background s-tile V which is adjacent to T through the right side of T to the actual tt-picture (that is, $\sharp$ of V is rewritten by a non-terminal). Although the tiles labeled by a and z from Figure 7 are exactly in the situation required by this rule, the rule is not applicable to these tiles. The reason is that the tile labeled by a is also adjacent to the tile labeled by t which belongs to M. The rule just described would be applicable only when the tile labeled by t would be also from the background. Informally, M cannot be grown by gluing tile z to tile a since the empty place indicated by tile z is (partly) occupied by tile t. Similarly, M cannot be grown by gluing the empty tile x to tile t since tile t is also adjacent to tile a.

We now prove that the family CF-ttPL properly includes the family Reg-ttPL, thus extending a corresponding result on array grammars generating arrays on the rectangular grid. We also show that there are tt-pictures, related to rectangular hollow frames, that cannot be generated by any CF-ttPG. This situation is again analogous to the case of arrays on rectangular grids [26].

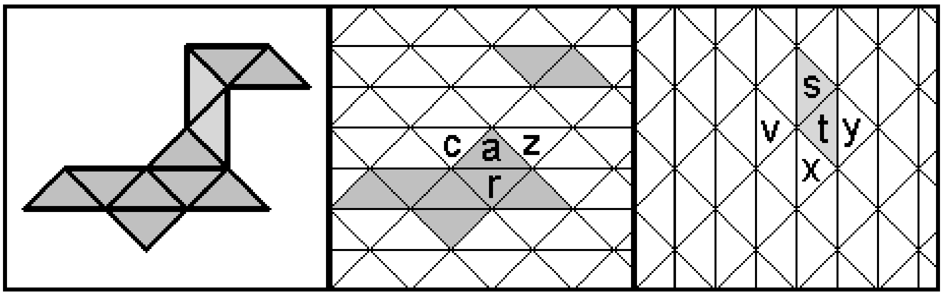

Fig. 7. A tt-picture and background tiles some of which here are labeled by symbols distinct from $\sharp$ only for explaining an example, see text.

Theorem 1. *The family of languages Reg-ttPL is a proper subset of CF-ttPL.*

Proof. The inclusion follows from the definitions as the rules of a Reg-ttPG are in fact special forms of the rules of a CF-ttPG. The proper inclusion can be seen by considering the language L_1 of tt-pictures of the shape of the alphabetic letter T, one member of which is shown in Figure 8. The language L_1 can be generated by a context-free tt-picture grammar $G = (N, \Sigma, P, (t, S), \sharp)$ where $N = \{S, A, B, C, D, E, X, Y\}$, $\Sigma = \{a\}$ and P consists of the rules shown in Figure 9. Note that only the first rule (containing the starting tile) is not regular.

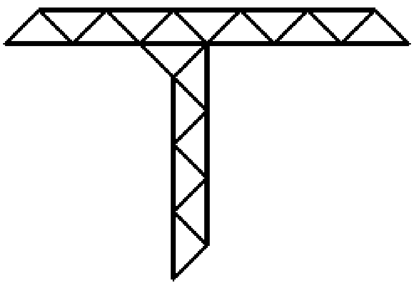

Fig. 8. T shaped tt-picture.

Starting with the n-tile (t, S) shown in Figure 9, an application of the only rule for (t, S) allows growing the three "arms" of the tt-picture in the shape of the letter T. This growth is done for the left arm by the rules which contain in their left hand sides a tile labeled by A or D and an empty tile. The right arm can be grown by the rules containing on the left a tile labeled by B or E and an empty tile. Finally, the rules containing on the left a tile labeled by C or X or Y permit to grow the vertical arm of the tt-picture in the shape of T.

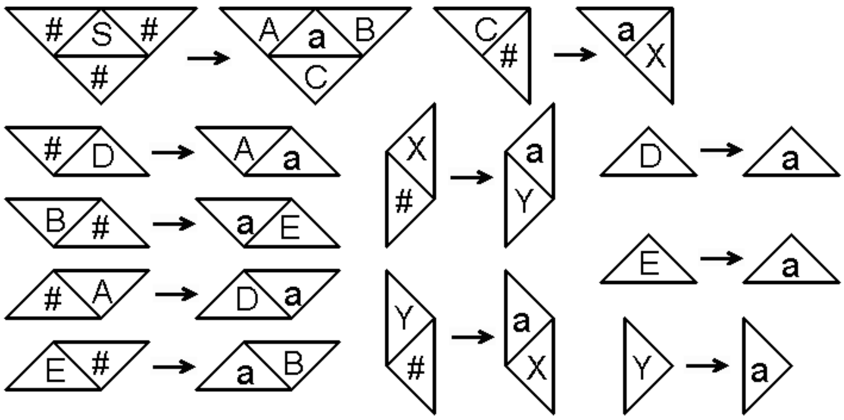

Fig. 9. Rules of a CF-ttPG for generating T shaped tt-pictures.

On the other hand, no regular tt-picture grammar can generate this. In a Reg-ttPG the rules cannot handle the "junction" of the three arms, in the sense that, if a Reg-ttPG begins generating, for example, the left arm and reaches the junction, it can either continue to generate the right arm or generate the vertical arm but once the junction is crossed, there is no way for the Reg-ttPG to return to generate the remaining arm. □

Theorem 2. *There exists no CF-ttPG which generates the language tt-RHF.*

Proof. If there is a CF-ttPG that can generate tt-RHF, then we can assume without loss of generality that there is a derivation in the grammar that starts from the "south-west" corner of the rectangular frame and grows nondeterministically the vertical arm (the left edge of the rectangular frame) and the horizontal arm (the bottom edge of the rectangular frame). At any time, the vertical arm can turn to the right, and then grow the upper edge of the rectangular frame; one further turning is possible, in the "north-east" corner of the rectangular frame. The computation has no control to stop at the "south-east" corner to yield the desired rectangular frame. In fact the computation can be terminated prior to crossing the bottom edge or after crossing it. □

4. Triangle-tiled picture P systems

It will be of more interest to consider tt-picture grammars in the framework of P systems, which we do in the following. The array-rewriting P system of Ceterchi et al [6] motivates the notion of a tt-picture P system which we define in this section and examine its properties in terms of its generative power. In a related study, making use of "iso-array grammars", P systems generating "iso-picture" languages are considered in [1] but again the same difficulty as in [12, 13] of having different kinds of blank symbols is present. Our definition of tt-picture P system overcomes this difficulty by allowing only one kind of blank symbol.

Definition 6. A *context-free triangle-tiled picture P system (CF-ttPP) of degree* $m \geq 1$ is a construct

$$\pi = (V, \Sigma, \sharp, \mu, F_1, \cdots, F_m, R_1, \cdots, R_m, i_0)$$

where

i) V is the total alphabet;
ii) $\Sigma \subset V$ is the terminal alphabet;
iii) $\sharp$ is the blank symbol;
iv) μ is a membrane structure with m membranes labeled 1 to m in a one-to-one manner;
v) $F_1, \cdots, F_m$ are finite sets of tt-pictures over V associated with the m membranes of μ (the tt-pictures of F_i belong to the membrane with index i);
vi) $R_1, \cdots, R_m$ are finite sets of context-free tt-picture grammar rules over V associated with the m membranes of μ (the rules of R_i are available only to be applied to the tt-pictures of F_i). Additionally, these rules have attached targets *here, out, in* (where in general, *here* is omitted).
vii) At least in one of the sets $F_1, \cdots, F_m$ there is an initial tt-picture.
viii) i_0 is the index of the output membrane.

In the special case that all rules in the sets $R_1, \cdots, R_m$ are regular tt-picture grammar rules over V, the CF-ttPP is named a *regular triangle-tiled picture P system (Reg-ttPP) of degree* $m \geq 1$.

In general, the definition of a P system has a complex hierarchical or nested membrane system where each membrane is allowed to contain one or more regions, combines grammars with grammar rules specified for each region, and the results of application of the rules are communicated between the regions of the membranes [18]. In the present work we use a simplified version of P system with a linear nested membrane structure where each membrane contains only one region. We represent the structure of m membranes as $\mu = [_1[_2 \cdots [_m]_m \cdots]_2]_1$, where the outermost (so-called skin) membrane has index 1 and contains a membrane inside which in turn contains another membrane inside and so on such that the innermost membrane has index m.

A ***computation*** in a CF-ttPP consists in performing an iterative process where in each iteration step a grammar rule is applied to one tt-picture (if applicable) from any region of the system, in analogy to the derivation in a CF-ttPG. We say also that in each step, a tt-picture is rewritten. So, the rewriting is sequential at the level of each tt-picture generated.

The result of a computation step performed by the application of a rule is a tt-picture (rewritten) which is placed in the region of the membrane indicated by the target associated with the rule as follows:

- If the rule has target *here* then the resulting tt-picture remains in the same region of the membrane.

- The target *out* means that the resulting tt-picture exits the current membrane and is sent to the immediately next outer membrane, that is, to the membrane with the next lower index. If the current membrane is the skin membrane (then no outer membrane does exist) then the computation result gets lost.

- The target *in* means that the resulting tt-picture is sent to one of the immediately inner membranes, non-deterministically chosen if several such membranes exist. If no internal membrane exists, then a rule with the target indication *in* cannot be used (the rule is not applicable).

Now, how the generation of tt-pictures is performed using a CF-ttPP? By the definition there is at least one initial tt-picture which belongs to the set F_i for some index $i \in \{1, \cdots, m\}$. If F_i lies in the output membrane ($i = i_0$) or if no rule of R_i is applicable to the initial tt-picture then the process is stopped. Otherwise the tt-picture is rewritten by some rule of R_i and the result is sent to some set F_j due to the target of that rule (for example, $j = i$ for the target *here*). Now the resulting tt-picture is treated in F_j, probably it is rewritten and sent to some membrane and so on. Probably several tt-pictures are generated "at the same time" by the CF-ttPP; although for each tt-picture generated, its generation process is sequential.

A ***computation is successful*** only if it stops and a so-called ***halting configuration*** is reached where no rule can be applied to the existing tt-pictures in all the sets $F_1, \cdots, F_m$, that is, no rule from R_i is applicable to all tt-pictures from F_i for all $i \in \{1, \cdots, m\}$. The ***result*** of such a successful computation consists of the tt-pictures labelled only by symbols from the terminal alphabet Σ and placed in the output membrane having index i_0. If a computation was successful and in the halting configuration the output membrane is empty then this computation has no result.

The set or language of all tt-pictures computed or generated as the result of a successful computation by a CF-tt-picture P system π is denoted by **CF-ttPL(π)**. The family of all languages CF-ttPL(π) generated by any CF-tt-picture P system π having degree at most m (with at most m membranes) is denoted by **CF-ttPPL(m)**. In analogy, the family of all languages Reg-ttPL(π) generated by any Reg-tt-picture P system π with at most m membranes is denoted by **Reg-ttPPL(m)**.

From the definitions it is clear that any CF-ttPP with only one membrane is a CF-ttPG, and conversely, any CF-ttPG can be seen as a CF-ttPP with one membrane. Analogously, any Reg-ttPP with one membrane is equivalent to a Reg-ttPG. We have seen above that for the corresponding language families we have that Reg-ttPL is a proper subset of CF-ttPL. In consequence,

$$\text{Reg-ttPPL(1) is a proper subset of CF-ttPPL(1)} \ .$$

The more general inclusion

$$\text{Reg-ttPPL(m)} \ \subseteq \ \text{CF-ttPPL(m)}$$

is evidently true since regular rules are special types of CF-rules. Nevertheless, the problem whether this inclusion is proper in general or for specific values m, remains open. The following theorem establishes that regular tt-picture P systems with three membranes has strictly more generation power than regular tt-picture grammars.

Theorem 3. *The family Reg-ttPL is a proper subfamily of Reg-ttPPL(3).*

Proof. We will give an example of a language L_1 which belongs to the set (Reg-ttPPL(3) $\setminus$ Reg-ttPL). Let L_1 be the set of all tt-pictures in the shape of V as in Figure 10(a) with the "arms" of V of equal length. L_1 cannot be generated by any regular tt-picture grammar since such a grammar can manage only one "growing head" at a time which means that the arms need not grow in equal length. But the language L_1 is generated by the Reg-tt-picture P system π_1 of degree three, given by the following specifications:

The membrane structure is given by $[_1[_2[_3]_3]_2]_1$, membrane 3 corresponds to the innermost membrane.

There is one initial tt-picture in membrane (with index) 1 which is shown in Figure 10(b). There are no initial pictures in other membranes.

The rule sets are shown in Figure 11. The rules r_1, r_2 are in membrane 1, r_2, r_4, r_5 in membrane 2 and r_6, r_7 in membrane 3. The target *in* means that a result has to be sent to the membrane with subsequently larger index. The target of each of the two rules in R_3 is *here* and therefore it is not mentioned in Figure 11.

The unique terminal symbol of π_1 is a and the non-terminals are known from the rules shown in Figure 11.

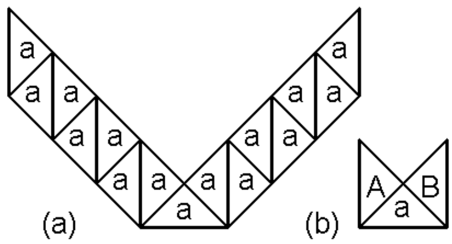

Fig. 10. (a) A tt-picture in the shape of V; (b) Initial tt-picture in membrane 1 of the Reg-tt-picture P system π_1.

The application of the rule r_1 to the initial tt-picture in membrane 1 causes growing its left arm one step. The resulting tt-picture is sent to membrane 2 due to the associated target *in*. Then, in membrane 2, the application of the rule r_2 sends it back to membrane 1, after growing the right arm one step. In membrane 1, the rule r_3 alone is applicable which grows the left arm again one step after which the resulting tt-picture is sent to membrane 2. In membrane 2, the rule r_2 is applicable which grows the right arm again one step after which the result is sent to membrane 1. The process repeats until an application of rule r_5 (instead of rule r_4) in membrane 2 sends the tt-picture to membrane 3 due to the target *in* of rule r_5. Now, in membrane 3, the application of rules r_6, r_7 terminates the computation yielding a tt-picture in the shape of V as in Figure 10(a). □

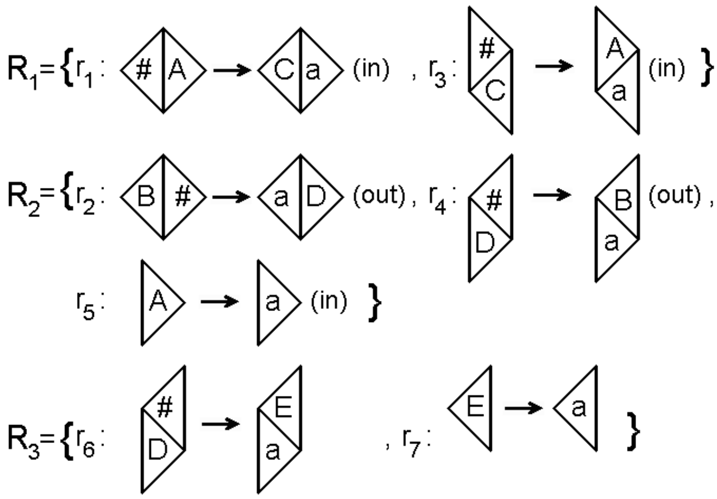

Fig. 11 Rules sets of the Reg-tt-picture P system π_1.

5. Triangle-tiled picture P systems and t-communication

Motivated by the study of [7], Subramanian et al [23] linked the cooperating array grammar systems [8] and the array P systems [6] through the t-communication mode. Here we extend this feature to tt-pictures by incorporating the t-communication mode into the tt-picture P systems. We describe a tt-picture P system with one type of t-communication mode, known as type *tin*.

Definition 7. *A t-communicating triangle-tiled picture P system of type* tin *is* given as a structure

$$\pi = (V, T, \sharp, \mu, F_1, \cdots, F_m, R_1, \cdots, R_m, i_0)$$

where the components are as in a corresponding context-free tt-picture P system (CF-ttPP) or regular tt-picture P system (Reg-ttPP).

A computation is as in a CF-ttPP except for the following differences:
A t-communicating CF-ttPP of type *tin* has rules in its membranes with target indication *out* or no target indication (and does not have rules with target indication *in*). If a rule has no target indication, then the tt-picture M to which the rule was applied remains in the same membrane if it can be further rewritten there. But if no rule can be applied to M in that membrane, then it is sent to the immediately next inner membrane if such a membrane exists. In other words the t-mode (which is also

called maximal derivation mode) enforces the target *in*. If the actual membrane is the output membrane, all rewritten tt-pictures remain there.

For a t-communicating triangle-tiled picture P system of type *tin*, the concepts of a successful computation, of a halting configuration and of the result of the computation are defined analogously as for a CF-ttPP.

The following theorem illustrates the power of a t-communicating tt-picture P system of type *tin* for generating the language tt-RHF of rectangular tt-hollow-frames (see Definition 2 and Figure 4). It shows that the special membrane system is used as control mechanism over the computation. Whereas there is no CF-ttPG which generates the language tt-RHF, it can be generated by a t-communicating tt-picture P system of type *tin* which even only uses regular rules.

Theorem 4. The language tt-RHF can be generated by a t-communicating tt-picture P system of type *tin*, with two membranes and regular grammar rules.

Proof. The language tt-RHF of rectangular tt-hollow-frames can be generated by a t-communicating tt-picture P system of type *tin* with a membrane structure given by $[_1[_2]_2]_1$ and specified as follows: The rule sets R_1, R_2 shown in Figure 12 correspond respectively to the membranes 1 and 2. The target attached with each of the rules in both sets R_1, R_2 is *here* and hence is not mentioned in the figure. The membrane 2 is the output membrane. The initial tt-picture belongs to the membrane 1 as shown in Figure 12 and there is no initial tt-picture in membrane 2. The terminal alphabet is $\{a\}$. The labels of the tiles in the rules shown in Figure 12 constitute the total alphabet.

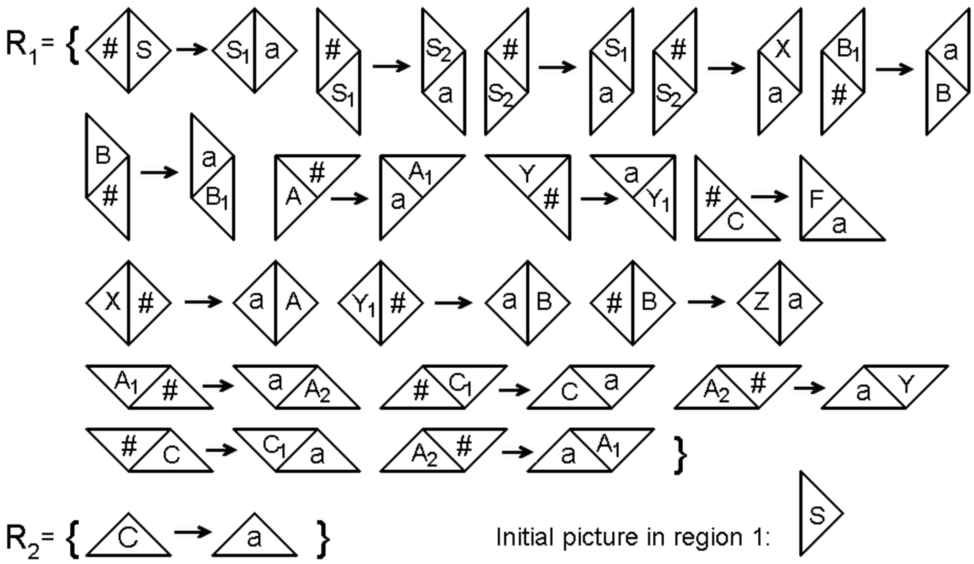

Fig. 12. Rule sets R_1, R_2 of the t-communicating tt-picture P system of type *tin* and the initial tt-picture in membrane 1.

Starting with the initial picture in membrane 1 and the application of the rules of R_1, the vertical left arm of a hollow rectangular frame (recall Figure 4) is grown followed by the upper horizontal arm and then the right vertical arm and finally the lower horizontal arm. An incorrect application of a rule for a tile with label C will lead to the derivation getting blocked with a tile with label F appearing in the tt-picture. But a correct sequence of application of the rules in the membrane 1 will send the tt-picture to membrane 2 due to the *tin* type of the system. Then, in membrane 2, an application of the only rule for C yields the desired rectangular tt-hollow-frame.$\square$

Remark: The notion of t-communication in P systems initially introduced in [7] and subsequently applied to (rectangular) array P systems in [23] has proved to be useful for the generation of rectangular frames tiled by isosceles right triangles labelled by a single symbol a. In [2] this problem is addressed but the approach considered here has two advantages over [2]: In our study the symbol $\sharp$ indicates an empty tile unlike in [2] wherein four kinds of blank symbols are used. In the rectangular frame considered in Theorem 9 a single symbol a is used to label each triangle-tile unlike in [2] wherein four terminal labels are used. So, our description is simpler.

5. Conclusion

Context-free triangle-tiled picture grammars (CF-ttPG) are studied here and some of their properties are obtained. Generating triangle-tiled pictures is also considered in the framework of context-free tt-picture P systems and particularly by t-communicating tt-picture P systems of type *tin*. Such P systems use a nested membrane system which is used as control mechanism over the computation. Therefore several figures can be generated by context-free tt-picture P systems which cannot be generated by any CF-ttPG. We prove in particular that there is no CF-ttPG which generates the language of rectangular hollow frames constructed by triangle-tiles, but that this language can be generated by a t-communicating tt-picture P system of type *tin* which even only uses regular rules. As for future work, the t-communication considered can be further explored by examining the *tout* type of t-communication introduced in [7]. The potential of the computation model investigated here for possible applications remains to be explored in relation to problems such as character recognition.

Acknowledgements. A preliminary version of the paper was presented at the 13th International Workshop on Combinatorial Image Analysis (IWCIA'2009). The first, third and fourth authors gratefully acknowledge support for this research from a FRGS grant No. 203/PKOMP/6711267 of the Ministry of Higher Education, Malaysia. The second author is thankful to CONACYT Mexico for support through the investigation project No. CB-2011-01-166223. The authors very gratefully acknowledge the detailed comments of the reviewers with several interesting and constructive remarks which helped to improve the presentation of the paper.

References

[1] Annadurai S., Thomas D. G., Dare V. R., Kalyani T., *Rewriting P systems generating iso picture languages*, LNCS 4958, Springer, Berlin Heidelberg, 2008, pp. 352–362.

[2] Annadurai S., Thomas D. G., Dare V. R., Kalyani T., *t-communicating iso-array P system*, *Proc. World Congress on Nature and Biologically Inspired Computing (NaBIC)*, 2009, pp. 1556–1559.

[3] Beauquier D., Nivat M., *Tiling the plane with one tile*, in: *Topology and Category Theory in Computer Science*, Oxford Univ. Press, Oxford, 1991, pp. 291–334.

[4] Beauquier D., Nivat M., *On translating one polyomino to tile the plane*, Discrete Comput. Geom., **6** (1991), pp. 575–592.

[5] Beauquier D., Nivat M., Remila E., Robson M., *Tiling figures of the plane with two bars*, Comput. Geom., **5** (1995), pp. 1–25.

[6] Ceterchi R., Mutyam M., Păun Gh., Subramanian K. G., *Array-rewriting P systems*, Natural Computing, **2** (2003), pp. 229–249.

[7] Csuhaj-Varju E., Vaszil G., Păun Gh., *Grammar systems versus membrane computing: The case of CD grammar systems*, Fundamenta Informaticae, **76** (2007), pp. 271–292.

[8] Dassow J., Freund R., Păun Gh., *Cooperating array grammar systems*, Int. Journal of Pattern Recognition and Artificial Intelligence, **9**(1995), pp. 1–25.

[9] Giammarresi D., Restivo A., *Two-dimensional languages*, in: *Handbook of Formal Languages*, Vol. **3**, (Eds. G. Rozenberg and A. Salomaa), Springer, 1997, pp. 215–267.

[10] Golomb S. W., *Tiling with polyominoes*, Journal of Combinatorial Theory, **1**(2)(1966), pp. 280–296.

[11] Grünbaum B., Shephard G. C., *Tilings and Patterns*, *A Series of Books in the Mathematical Sciences*, W. H. Freeman and Company, New York, 1989.

[12] Kalyani T., Dare V. R., Thomas D. G., *Iso-array grammars and picture languages*, Int. J. Math. Sci., **6**(2007), no. 3–4, pp. 369–384.

[13] Kalyani T., Dare V. R., Thomas D. G., *Recognizability of iso-picture languages by Wang systems*, Intern. J. Imaging Systems Tech., **19**(2009), pp. 140–145.

[14] Kalyani T., Sasikala K., Dare V. R., Abisha P. J., Robinson T., *Triangular pasting system*, in: *Formal Models, Languages and Application*, (Eds. K. G. Subramanian, K. Rangarajan, M. Mukund), Series in Machine Perception and Artificial Intelligence, **66** (2006), pp. 195–211.

[15] Laroche P., Nivat M., Saoudi A., *Context-sensitivity of puzzle grammars*, LNCS 654, Springer, 1992, pp. 195–212.

[16] Nivat M., Saoudi A., Subramanian K. G., Siromoney R., Dare V. R., *Puzzle grammars and context-free array grammars*, International Journal of Pattern Recogn. Artificial Intell., **5**(1991), pp. 663–676.

[17] Ollinger N., *Tiling the plane with a fixed number of polyominoes*, *Proc. of the Third Int. Conf. on Language and Automata Theory and Applications (LATA'2009)*, (Eds. A. H. Dediu, A. M. Ionescu, C. Martin-Vide), LNCS 5457, Springer, Berlin, 2009, pp. 638–647.

[18] Păun Gh., *Computing with membranes*, J. Comp. System Science, **61**(2000), pp. 108–143.

[19] Robinson T., Dare V. R., Subramanian K. G., *A parallel pattern generating system*, *Proc. of the 6th Int. Workshop on Parallel Image Processing and Analysis*, 1999, pp. 23–32.

[20] Rosenfeld A., *Picture Languages – Formal Models for Picture Recognition*, Academic Press, New York, 1979.

[21] Rosenfeld A., Siromoney R., *Picture languages – a survey, Languages of Design*, **1**(1993), pp. 229–245.

[22] Subramanian K. G., *P systems and picture languages*, LNCS 4664, Springer, 2007, pp. 99–109.

[23] Subramanian K. G., Ali R. M., Nagar A. K., Margernstern M., *Array P systems and t-communication*, Fundam. Inform., **91**(2009), pp. 145–159.

[24] Subramanian K. G., Geethalakshmi M., Nagar A. K., Lee S. K., *Triangle-tiled picture languages*, in: *Progress in Combinatorial Image Analysis*, (Eds. P. Wiederhold, R.P. Barneva), Research Publishing Services, 2010, pp. 165–180.

[25] Wang P. S. P., *Array Grammars, Patterns and Recognizers*, World Scienfic, Singapore, 1989.

[26] Yamamoto Y., Morita K., Sugata K., *Context-sensitivity of two-dimensional regular array grammars*, in: *Array Grammars, Patterns and Recognizers*, (Ed. P.S.P. Wang), WSP Series in Computer Science, **18**, World Scientific Publ., Singapore, 1989, pp. 17–41.