Original Paper ______ *Forma*, **22**, 47–54, 2007

P Systems for Array Generation and Application to Kolam Patterns

K. G. SUBRAMANIAN[1]*, R. SARAVANAN[2] and T. ROBINSON[3]

[1]*School of Mathematical Sciences, University Sains Malaysia, 11800 Penang, Malaysia*
[2]*Department of Mathematics, Bharath Institute of Higher Education and Research, Chennai 600073, India*
[3]*Department of Mathematics, Madras Christian College, Chennai 600059, India*
**E-mail address: kgs@usm.my*

(Received November 3, 2006; Accepted September 5, 2007)

Keywords: Kolam Patterns, Array Language, P-system

Abstract. In the area of membrane computing, a new computability model, now called P system was introduced by PAUN (2002) inspired from the cell structure and its functioning. One area of P systems deals with string objects and rewriting rules. Recently, array P systems with array objects and array rewriting rules were introduced. Here we introduce a new class of array P systems called sequential/parallel rectangular array P systems generating pictures of rectangular arrays. These P systems have rectangular arrays as objects in the membranes and rules in the membranes are either context-free or regular with sequential horizontal rewriting or right-linear rules with vertical rewriting in parallel as in the two-dimensional (2D) matrix grammars (SIROMONEY *et al*., 1972). These P systems have more generative power and accordingly, they can generate kolam patterns that cannot be handled by 2D matrix grammars.

1. Introduction

In syntactic approaches to generation of picture patterns, several two-dimensional grammars have been proposed. One of the earliest models was proposed by SIROMONEY *et al*. (1972), motivated by "kolam" patterns (SIROMONEY *et al*., 1974). In this model, now called as two-dimensional (2D) matrix grammar, generation of rectangular arrays takes place in two phases with a sequential mode of rewriting in the first phase generating strings of intermediate symbols and a parallel mode of rewriting these strings in the second phase to yield rectangular picture patterns. Applications of these models to generation of "kolam" patterns have also been made in the literature (SIROMONEY *et al*., 1974).

On the other hand a new computability model in the area of membrane computing, now called P system, was introduced by PAUN (2002), inspired from the cell structure and its functioning. This model has been investigated by several researchers and is a promising framework for many problems. The basic elements of a P system are the membrane structure and evolution rules which process objects in the compartments of the membrane structure. Each membrane uniquely determines a compartment, also called region, the space delimited from above by it and from below by the membranes placed directly inside, if any exist. There is an external membrane called the skin membrane and there may be several internal membranes. In the basic class of P systems, each region contains objects that evolve by means of evolution rules associated with the regions. There could be communication of objects through membranes. The used objects are “consumed”, the newly produced objects are placed in the compartments according to the communication commands assigned to them. The rules to be used and the objects to evolve are chosen in a non-deterministic manner. All compartments of the system evolve at the same time, synchronously. Among several areas of study of P systems, one area deals with P systems with string objects and string rewriting rules. In recent years, study on relating array rewriting systems generating picture languages and P systems has been made by CETERCHI *et al.* (2003) by considering array objects and array rewriting rules.

Here we introduce a new class of array P systems called sequential/parallel rectangular array P systems generating pictures of rectangular arrays. These P systems have rectangular arrays as objects in the membranes and rules in the membranes are either context-free or regular with sequential horizontal rewriting or right-linear rules with vertical rewriting in parallel as in the 2D matrix grammars. When compared with the 2D matrix grammars these P systems have more generative power and accordingly, they can generate kolam patterns that cannot be handled by two-dimensional matrix grammars.

2. Basic Definitions

For notions and notations relating to arrays we refer to SIROMONEY *et al.* (1972). Let Σ be a finite alphabet. Σ^* is the set of all words over Σ including the empty word. A picture A over Σ is a rectangular $m \times n$ array of elements of Σ. The set of all pictures over Σ is denoted by Σ^{**}. A picture language or a two dimensional language over Σ is a subset of Σ^{**}. The column concatenation $A\Phi B$ of an $m \times p$ array A and an $n \times q$ array B is defined only when $m = n$ and the row concatenation $A\theta B$ of A and B is defined only when $p = q$. We now recall the two-dimensional (2D) matrix grammars of SIROMONEY *et al.* (1972).

Definition 1 (Two-Dimensional (2D) Matrix Grammars)

A 2D matrix grammar is a 2-tuple (G_1, G_2) where $G_1 = (H_1, I_1, P_1, S)$ is a Regular, CF or CS grammar; H_1 is a finite set of symbols called horizontal nonterminals; $I_1 = \{S_1, S_2, ..., S_k\}$, a finite set of symbols called intermediates; $H_1 \cap I_1 = \phi$; P_1 is a finite set of rules called horizontal rules; S is the start symbol; $S \in H_1$; $G_2 = (G_{21}, G_{22}, ..., G_{2k})$ where $G_{2i} = (V_{2i}, T, P_{2i}, S_i)$, $1 \le i \le k$ are regular grammars; G_{2i} is a finite set of symbols called vertical nonterminals; $V_{2i} \cap V_{2j} = \phi$ for $i \neq j$; T is a finite set of terminals; P_{2i} is a finite set of right-linear rules of the form $X \to aY$ (called vertical nonterminal rules) or $X \to a$ (called vertical terminal rules) where $X, Y \in V_{2i}$, $a \in T$; $Si \in V_{2i}$ is the start symbol of G_{2i}.

G is a regular, (context-free, context-sensitive) 2D matrix grammar if G_1 is regular, (context-free, context sensitive) respectively. Derivations are defined as follows: First a string w over I_1 is generated horizontally using the horizontal rules. Vertical derivations then proceed in parallel using the rules of G_{2i}. All symbols in the horizontal string w are rewritten in the vertical direction using vertical nonterminal rules for all symbols of w or vertical terminal rules for all symbols of w, thus generating rectangular arrays over T when the vertical derivation terminates.

The set L(G) consists of all $m \times n$ arrays generated by G. We denote the picture language classes of regular, CF, CS 2DMatrix grammars by 2DRML, 2DCFML, 2DCSML respectively.

3. Sequential/Parallel Array P Systems

Here we introduce a new kind of array P systems by having the objects in the membranes as rectangular arrays and the rules as either context-free or regular rules or sets of vertical nonterminal/terminal rules as considered in a 2D matrix grammar.

Definition 2

A sequential/parallel Rectangular Array P system of degree m (≥1) is a construct

$$\pi = (V_1 \cup V_2, I, T, \mu, F_1, ..., F_m, R_1, ..., R_m, i_0)$$

where $V = V_1 \cup V_2$ is the total alphabet; V_1-I is the set of horizontal nonterminals; $I \subset V_1$ is the set of intermediates; V_2-T is the set of vertical nonterminals; $T \subseteq V_2$ is the set of terminals; V_2-T includes the elements of I; m is a membrane structure with m membranes labeled in a one-to-one way with 1, 2, ..., m; $F_1, ..., F_m$ are finite sets of rectangular arrays over V associated with the m regions of μ; $R_1, ..., R_m$ are finite sets of rules associated with the m regions of μ; the rules can be either horizontal context free rules of the form $A \to \alpha$, $A \in V_1$-I, $\alpha \in V_1^*$ or a set of right linear vertical nonterminal rules of the form $X \to aY$, $X, Y \in V_2$-T, $a \in T$ or a set of right-linear vertical terminal rules of the form $X \to a$, $X \in V_2$-T, $a \in T$.

The horizontal CF rules can be in particular regular rules of the form $A \to wB$, $A \to w$, $A \in V_1$-I, $w \in I^*$. Horizontal rules and sets of vertical rules have attached targets, *here*, *out*, *in* (in general, *here* is omitted). A membrane has either horizontal rules or sets of vertical rules; horizontal rules are applied in a sequential manner; the vertical rules in a parallel manner in the vertical direction as in a 2D matrix grammar. Finally, i_0 is the label of an elementary membrane of m (the out put membrane).

A computation in a sequential/parallel array P system is defined in the same way as in a string rewriting P system with the successful computations being the halting ones; each rectangular array in each region, which can be rewritten by a horizontal rule or a set of vertical rules associated with that region (membrane), should be rewritten; the rectangular array obtained by rewriting is placed in the region indicated by the target associated with the rule used (*here* means that the array remains in the same region, *out* means that the array exits the current membrane and thus, if the rewriting was done in the skin membrane, then it can exit the system; arrays leaving the system are "lost" in the environment, and *in* means that the array is immediately sent to one of the directly lower membranes, non deterministically chosen if several exist; if no internal membrane exists, then a rule with the target indication *in* cannot be used).

A computation is successful only if it stops; a configuration is reached where no rule can be applied to the existing arrays. The result of a halting computation consists of the rectangular arrays composed only of symbols from T placed in the membrane with label i_0

$$\begin{array}{ccccccc} a & a & a & a & a & a & a \\ b & b & b & a & b & b & b \\ b & b & b & a & b & b & b \\ b & b & b & a & b & b & b \\ b & b & b & a & b & b & b \end{array} \qquad \begin{array}{ccccccc} a & a & a & a & a & a & a \\ & & & a & & & \\ & & & a & & & \\ & & & a & & & \\ & & & a & & & \end{array}$$

Fig. 1. Fig. 2.

Fig. 1. Array of Token T.
Fig. 2. Token T of a's.

in the halting configuration.

The set of all such arrays computed (we also say generated) by a system Π is denoted by RAL (Π). The family of all array languages RAL (Π). generated by systems Π as above, with at most m membranes, with horizontal rules of type $\alpha \in$ {REG, CF} is denoted by S/PRAP$_m$ (α).

Example 1.

Consider the Sequential/Parallel Rectangular Array P system belonging to the class S/PRAP$_4$ (REG)

$\Pi_1 = (V_1 \cup V_2, I, T, [_1[_2[_3[_4]_4]_3]_2]_1, AS_2B, \varphi, \varphi, \varphi, R_1, R_2, R_3, R_4, 4)$
$V_1 = \{A, B, C, S_1, S_2\}$, $V_2 = \{S_1, S_2, D, a, b\}$, $I = \{S_1, S_2\}$, $T = \{a, b\}$
$R_1 = \{A \to S_1A\ (in)\}$
$R_2 = \{B \to S_1B\ (out), B \to S_1C\ (in)\}$
$R_3 = \{A \to S_1\ (here), C \to S_1\ (in)\}$
$R_4 = \{\{S_1 \to \begin{smallmatrix} a \\ D \end{smallmatrix}, S_2 \to \begin{smallmatrix} a \\ S_2 \end{smallmatrix}\}\ (here), \{D \to \begin{smallmatrix} b \\ D \end{smallmatrix}, S_2 \to \begin{smallmatrix} a \\ S_2 \end{smallmatrix}\}\ (here), \{D \to b, S_2 \to a\}\ (here)\}$.

The array object AS_2B (which is indeed a string) is initially in the membrane 1 and the other membranes do not have objects. The rule $A \to S_1\ A$ is applied to AS_2B to yield S_1AS_2B, which is sent to the inner membrane 2. If the rule $B \to S_1\ B$ is applied in membrane 2, $S_1AS_2S_1B$ is generated and sent back to membrane 1 as the target attached to the rule is *out* and the process repeats but if the rule used is $B \to S_1C$ with target *in*, then $S_1AS_2S_1C$ is sent to inner membrane 3 wherein $S_1S_1S_2S_1S_1$ is generated and sent to inner membrane 4. The process can be repeated to generate $S_1{}^nS_2S_1{}^n$. In membrane 4 the first of the three sets of vertical rules is applied followed by the second, a certain number of times, the computation halting with the application of the third of the sets of vertical rules. One of the rectangular arrays generated is shown in Fig. 1. Any premature application of $C \to S_1$ in membrane 3 sends $S_1AS_2S_1S_1$ to membrane 4 where it gets stuck as the sets of vertical rules do not have a rule for A.

The picture language generated by Π_1 consists of rectangular arrays over a (Fig. 1) describing token T (Fig. 2) with equal "horizontal arms" when b is interpreted as blank.

Now we examine the generative power of these Sequential/parallel rectangular array P systems.

Theorem 1.

i) S/P RAP_4 (REG) $\supset$ RML

ii) S/P RAP_4 (CFG) $\cap$ CFML $\neq \phi$

Proof

The proper inclusion in statement (i) follows from Example 1, as no RMG can generate picture arrays describing token T as in Fig. 1 since the horizontal "arms" of Token T are equal in length.

Given a RMG, G = (G_1,G_2) the inclusion in (i) is seen by constructing a sequential/parallel rectangular array P system with just 2 membranes. The membrane structure is $[_1[_2]_2]_1$. Initially the start symbol S of G_1 is in membrane 1 and there are no objects in 2. The rules in membrane 1 are the horizontal regular nonterminal rules with target *here* and horizontal regular terminal rules, with target *in*. The rules in membrane 2 are of two kinds: a set consisting of all vertical right linear nonterminal rules of G_2 with the target *here* and another set consisting of all vertical right linear terminal rules of G_2 with target *here*. It can be seen that this array P system generates the picture language generated by G.

The proof of ii) is due to the fact that the set of arrays describing token T of *a*'s with equal horizontal "arms" can indeed be generated by a CF 2Dmatrix grammar (SIROMONEY *et al.*, 1972).

Theorem 2.

S/$PRAP_4$ (REG) $\subset$ S/$PRAP_4$ (CFG)

Proof

The inclusion is clear from the definitions. The proper inclusion can be seen as follows: Consider the Sequential/Parallel Rectangular Array P system in the class S/$PRAP_4$ (CFG)

$\Pi_2 = (V_1 \cup V_2, I, T, [_1[_2[_3[_4]_4]_3]_2]_1, AC, \varphi, \varphi, \varphi, R_1, R_2, R_3, R_4, 4)$
$V_1 = \{A, C, S_1, S_2, S_3\}$, $V_2 = \{S_1, S_2, S_3, a, b, c\}$, $I = \{S_1, S_2, S_3\}$, $T = \{a, b, c\}$
$R_1 = \{A \to S_1AS_2$ (*in*),$\}$
$R_2 = \{C \to S_3C$ (*out*), $C \to S_3D$ (*in*)$\}$
$R_3 = \{A \to S_1S_2$ (*here*), $D \to S_3$ (*in*)$\}$
$R_4 = \{\{S_1 \to a, S_2 \to b, S_2 \to c\}$ (*here*), $\{S_1 \to a, S_2 \to b, S_3 \to c\}$ (*here*).
S_1 S_2 S_3

The array object (in fact a string) AC is initially in membrane 1 with other membranes being empty. The rule $A \to S_1AS_2$ generates S_1AS_2C which is sent to membrane 2 wherein the application of $C \to S_3D$ generates $S_1AS_2S_3D$ which is then sent to membrane 3. Here the string is changed to $S_1S_1S_2S_2S_3S_3$. On the other hand if the rule $C \to S_3C$ is applied instead of $C \to S_3D$, then $S_1AS_2S_3C$ is sent back to membrane and the process can be repeated. Ultimately strings of the form $S_1{}^nS_2{}^nS_3{}^n$ are generated in membrane 3. These

$$\begin{array}{cccccccccccc} a & a & a & a & b & b & b & b & c & c & c & c \\ a & a & a & a & b & b & b & b & c & c & c & c \\ a & a & a & a & b & b & b & b & c & c & c & c \\ a & a & a & a & b & b & b & b & c & c & c & c \\ a & a & a & a & b & b & b & b & c & c & c & c \end{array}$$

Fig. 3. Array of three equal size blocks of a's, b's, c's.

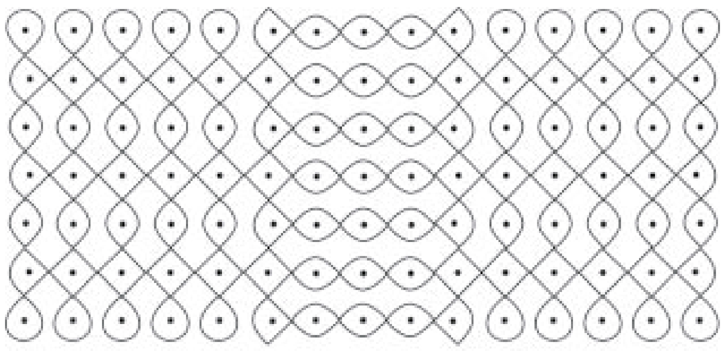

Fig. 4. A P System Kolam.

strings are sent to membrane 4 wherein pictures as in Fig. 2 are generated.

The picture language generated by Π_2 consists of rectangular arrays of three equal size blocks of a's, b's, c's (Fig. 2).

4. Application to "Kolam" Pattern Generation

"Kolam" refers to decorative artwork drawn on the floor with the kolam drawing generally starting with a certain number pattern of points and curly lines going around these points. Classification of kolam patterns based on their generation by different array grammars has been considered by SIROMONEY *et al*. (1974). The Sequential/parallel rectangular array P systems introduced here, being more powerful than the 2D matrix grammars, are suitable in generating kolam patterns that cannot be generated by regular 2D matrix grammars. The approach for generation of kolam patterns adopts a technique referred to as Narasimhan's method of kolam generation (SIROMONEY *et al*., 1974). The kolam patterns are coded as rectangular arrays of symbols. These arrays are generated using the P systems introduced here and then substitution of the basic units of the kolam pattern takes place yielding the desired patterns.

As an illustration we consider the kolam pattern in Fig. 3.

The kolam pattern in Fig. 3 can be expressed as an array (Fig. 5) using primitive patterns by NAGATA and ROBINSON (2006).

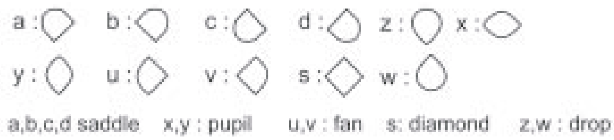

Fig. 5. Primitive patterns of the P System Kolam.

$$
\begin{array}{ccccccccccccccc}
z & z & z & z & z & u & x & x & x & v & z & z & z & z & z \\
u & s & s & s & s & s & x & x & x & s & s & s & s & s & v \\
y & y & y & y & y & u & x & x & x & v & y & y & y & y & y \\
u & s & s & s & s & s & x & x & x & s & s & s & s & s & v \\
y & y & y & y & y & u & x & x & x & v & y & y & y & y & y \\
u & s & s & s & s & s & x & x & x & s & s & s & s & s & v \\
w & w & w & w & w & u & x & x & x & v & w & w & w & w & w
\end{array}
$$

Fig. 6. Array representation of kolam in Fig. 3.

The primitive patterns corresponding to the symbols (0 for arc line and 1 for linear line) in the array in Fig. 5 are as follows in the notation by NAGATA and ROBINSON (2006):

a, b, c, d: saddle 0011, 0110, 1001, 1100
x, y: pupil 0101, 1010
u, v: fan 1011, 1110
s: diamond 1111
z, w: drop 0010, 1000

The set of such kolam patterns can be generated by a Sequential/parallel rectangular array P system similar to the P system Π_2 in the proof of Theorem 2, with slight modifications but cannot be generated by any CF 2D matrix grammar as the "middle part" and the "left/right parts" have equal number of columns in the kolam.

5. Conclusion

We have introduced here a new type of array P system called S/P rectangular array P system based on 2D matrix grammars. We have exhibited the suitability of these systems for Kolam pattern generation. One of the problems that needs further study is the minimum number of membranes needed for generation of kolam patterns discussed here.

The authors would like to thank the referees for their useful comments. The first author K. G. Subramanian is grateful to Prof. Dr. Rosihan M. Ali for his interest and academic support and to Universiti Sains Malaysia for their financial support.

REFERENCES

CETERCHI, R., MUTYAM, M., PAUN, Gh. and SUBRAMANIAN, K. G. (2003) Array-rewriting P systems, *Natural Computing*, **2**, 229–249.

NAGATA, S. and ROBINSON, T. (2006) Digitalization of kolam patterns and tactile kolam tools, in *Formal Models, Languages and Applications* (K. G. Subramanian, K. Rangarajan and M. Mukund, eds.), Series in Machine Perception and Artificial Intelligence, Vol. 66, pp. 354–363.

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

SIROMONEY, G., SIROMONEY, R. and KRITHIVASAN, K. (1972) Abstract families of matrices and picture languages, *Computer Graphics and Image Processing*, **1**, 234–307.

SIROMONEY, G., SIROMONEY, R. and KRITHIVASAN, K. (1974) Array grammars and kolam, *Computer Graphics and Image Processing*, **3**, 63–82.