Generation of Kolam-Designs Based on Contextual Array P Systems

Ibrahim Venkat[1], Thamburaj Robinson[2], K. G. Subramanian[2(✉)], and Philippe de Wilde[3]

[1] School of Computer Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia
ibra@usm.my
[2] Department of Mathematics, Madras Christian College, Tambaram, Chennai 600059, India
robinson@mcc.edu.in, kgsmani1948@gmail.com
[3] University of Kent, Canterbury, UK
p.dewilde@kent.ac.uk

Abstract. Kolam-designs are diagrams used to decorate the floor, especially in front of a house in South India. Methods of generation of the kolam diagrams were developed based on two-dimensional picture generating models, broadly known as array grammars, introduced for the description and analysis of picture patterns. Rewriting array P system, a membrane computing model based on array rewriting has been developed to evolve picture arrays, based on context-free array rewriting rules. In contrast to this array P system, another P system model called contextual array P system (CAP) using contextual array rules for the evolution or generation of picture arrays has been proposed and its power in generating picture arrays investigated. Here we develop an application of CAP for the generation of the kolam diagrams. The advantage of using CAP is that kolam diagrams that cannot be handled by array grammars can be generated by the CAP model.

Keywords: Floor-designs · Kolam patterns · Array grammars
Array P system · Contextual array rules

1 Introduction

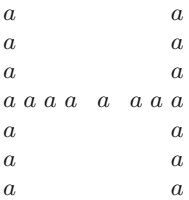
Kolam-designs are visually pleasing geometric diagrams that are drawn in the ground mostly at the entrance of a home [1,5], with this age-old tradition being more prevalent in South India. A kolam is generally drawn starting with a certain number of points arranged in a particular pattern, with curly lines being drawn around these points, resulting in the intended "kolam" drawing. In fact there are very intricate and complicated kolam patterns used by the kolam practitioners. Motivated by these "kolam" floor-designs Siromoney et al. [8], in the years 1972–74, proposed picture generating models, called array grammars, and described a technique [8], referred to as Narasimhan's method, for the generation

P. Chapman et al. (Eds.): Diagrams 2018, LNAI 10871, pp. 79–86, 2018.
https://doi.org/10.1007/978-3-319-91376-6_11

of kolam patterns using these array grammars. Subsequently, several researchers have developed a variety of picture-array models. These grammars make use of two-dimensional extension of Chomsky type of string grammars or contextual type of string grammars [7] in the area of formal languages. On the other hand a bio-inspired computing model with a generic name of P system was introduced by Paun [6] which has turned out to be a versatile model with applications in different areas, with the area of picture array generation being one among these application areas. Several P system models have been proposed for picture array generation [2,9]. Based on contextual type array generating rules [4], an array P system, called contextual array P system (CAP) has been introduced in [3]. Here we construct CAP for generation of kolam patterns that cannot be handled by contextual array grammars [4] utilizing the technique of Narasimhan's method of kolam generation.

2 Preliminaries

We recall needed notions and related results [3,4]. A point $p = (m, n)$ in the two-dimensional digital plane with integer coordinates m, n, is identified with a unit closed square and is called a pixel or cell. The pixels $p_1 = (m-1, n)$, $p_2 = (m+1, n)$, $p_3 = (m, n-1)$ and $p_4 = (m, n+1)$, are neighbours of the pixel $p = (m, n)$. Now let V be a finite set of symbols, referred to as an alphabet and # a symbol not belonging to V. A two-dimensional picture array (or simply called an array) over V is a finite set of connected pixels labeled by the symbols of V. It is sufficient for our purposes to give an array in pictorial form without mentioning the coordinates of the pixels. Note that by connected picture array we mean that every pixel labelled by a symbol of V has at least another labelled pixel of the picture array as a neighbour. The symbol # is used as a label of a pixel to indicate that it is empty *i.e.* it is not occupied by any label of V. When specifying a picture array we can list the coordinates of the pixels of the array with their labels in a formal manner but it is enough to specify the labels of the pixels of the array. We denote the set of all non-empty connected finite picture arrays over V by V^{++}. A subarray Y of an array X is a connected part of X. As an illustration, the picture array in Fig. 1(A) with each pixel having label a, denotes a digitized form of the letter H while in Fig. 1(B), a labelled square array is shown with $d_1, d_2, f_1, f_2, p_v, d$ being the labels of the pixels. A "kolam" pattern can be thought of as a picture array in the two-dimensional plane as described in [5] with the cells of the picture array labelled by primitive "kolam" patterns [5], a list of which as shown in Fig. 2 and an example "kolam" is shown in Fig. 3. We note that based on the "Narasimhan's method" of "kolam" generation described in [8], a "kolam" pattern, which is composed of primitive "kolam" patterns, can be transformed into a picture array over the symbols of the primitive patterns and conversely. As an example, the picture array in Fig. 1(B) corresponds to the "kolam" in Fig. 3. Nagata and Robinson [5] propose an interesting view of a "kolam" as being composed of primitive kolam patterns, a partial list of which is given in Fig. 2.



(A)

$$\begin{array}{ccccc} d_1 & d_1 & d_1 & d_1 & d_1 \\ f_1 & d & d & d & f_2 \\ p_v & p_v & p_v & p_v & p_v \\ f_1 & d & d & d & f_2 \\ d_2 & d_2 & d_2 & d_2 & d_2 \end{array}$$

(B)

Fig. 1. (A) An array representing the letter H; (B) a square array

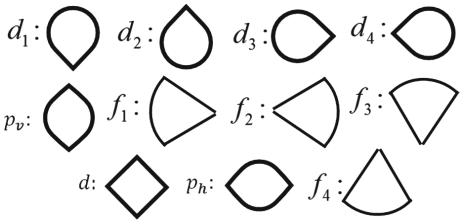


Fig. 2. Kolam primitive patterns.

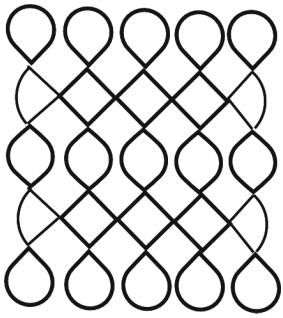


Fig. 3. A kolam pattern.

3 Contextual Array P System and "Kolam" Pattern Generation

In [3], contextual array P system is considered and several theoretical properties of this system are established. We first illustrate the manner of application of a rule of a contextual array grammar (CAG) [4] and then recall contextual array P system [3]. In a contextual array grammar (CAG) rule of the form $r : (\alpha, \beta)$, the arrays α and β are referred to as "selector" and "context" respectively. We restrict ourselves to the case with the "selector" and the "context" in the contextual array grammar rule not having empty pixels *i.e.* the "selector" and the "context" are connected and labeled only by symbols from an alphabet V and not by the blank symbol #. For arrays $X, Y \in V^{++}$ where V is the alphabet, intuitively, if in X, we find a subarray that corresponds to the selector α, and if the places (*i.e.* unit squares in the two-dimensional plane) corresponding to β are labeled only by the blank symbol #, then we can add the context β to α, thus deriving an array M and we write $X \Longrightarrow_G M$. If an array Z is obtained from an array X, by a sequence of such steps, we write $X \Longrightarrow_G^* Z$. A maximal mode or t-mode of derivation, denoted $\Longrightarrow_G^t$ corresponds to collecting only the arrays produced by blocked derivations, namely, derivations which cannot be continued. For a formal definition of CAG we refer to [4]. In the $*$-*mode* of derivation, all picture arrays derivable from an axiom are taken in the picture language (also called, array language) generated by G while in the t-mode of derivation, all

picture arrays produced by blocked derivations constitute the picture language. We give an example of a contextual array grammar working in t-mode generating the array language L_1 consisting of all picture arrays which are squares of side lengths $2n+3, n \geq 1$, with the label pixels of the picture arrays belonging to V. In such a picture array of L_1, the first row is of the form d_1^{2n+3}, the last row is of the form d_2^{2n+3} and every two adjacent rows starting from the second row are of the form $f_1 d^{2n+1} f_2$ and p_v^{2n+3} respectively. A 5×5 picture array of L_1 is shown in Fig. 1(B) and represents the "kolam" in Fig. 3. We consider a contextual array grammar G with an axiom picture array A_1, which is given in pictorial form: $A_1 := \begin{matrix} f_1 & & \\ d_2 & d_2 & \end{matrix}$ The contextual array rules of G are given by the following pictorial forms where the pixels and symbols of the selector are indicated by enclosing them in boxes. In other words the rules $p_i (1 \leq i \leq 17)$ are given as follows:

$$p_1 := \begin{matrix} \boxed{f_1} & p & \\ \boxed{d_2} & \boxed{d} & d \end{matrix}, p_2 := \begin{matrix} d & & \\ \boxed{p} & p & p \\ \boxed{d} & \boxed{d} & \end{matrix}, p_3 := \begin{matrix} p & & \\ \boxed{d} & d & d \\ \boxed{p} & \boxed{p} & \end{matrix}, p_4 := \begin{matrix} d_1 & & \\ \boxed{d} & d & f_2 \\ \boxed{p} & \boxed{p} & \end{matrix}, p_5 := \begin{matrix} \boxed{d_1} & d_1 \\ \boxed{d} & \boxed{f_2} \end{matrix}$$

$$p_6 := \begin{matrix} f_1 & \boxed{d} \\ \boxed{p} & \boxed{p} \\ \boxed{f_1} & \end{matrix}, \quad p_7 := \begin{matrix} p & \boxed{p} \\ \boxed{f_1} & \boxed{d} \end{matrix}, \quad p_8 := \begin{matrix} d & \boxed{d} \\ \boxed{p} & \boxed{p} \\ \boxed{d} & \end{matrix}, \quad p_9 := \begin{matrix} p & \boxed{p} \\ \boxed{d} & \boxed{d} \end{matrix}, \quad p_{10} := \begin{matrix} d_1 & \boxed{d_1} \\ \boxed{f_1} & \boxed{d} \end{matrix},$$

$$p_{11} := \begin{matrix} d_1 & \boxed{d_1} \\ \boxed{d} & \boxed{d} \end{matrix}, \quad p_{12} := \begin{matrix} \boxed{d} & \boxed{d} \\ \boxed{d_2} & d_2 \end{matrix}, \quad p_{13} := \begin{matrix} & \boxed{d} & \\ \boxed{p} & \boxed{p} & \\ \boxed{d} & d & \end{matrix}, \quad p_{14} := \begin{matrix} & \boxed{d} & \\ \boxed{p} & \boxed{p} & \\ \boxed{p} & p & \end{matrix}, \quad p_{15} := \begin{matrix} \boxed{d} & \boxed{f_2} \\ \boxed{d_2} & d_2 \end{matrix},$$

$$p_{16} := \begin{matrix} & \boxed{f_2} & \\ \boxed{p} & \boxed{p} & \\ \boxed{d} & f_2 & \end{matrix}, \quad p_{17} := \begin{matrix} \boxed{d} & \boxed{f_2} \\ \boxed{p} & p \end{matrix}$$

A maximal (that is, t-mode) derivation in G generating a 5×5 picture array of the language L_1 is done by the application of the rules $p_1, p_2, p_4, p_5, p_7, p_8, p_6, p_{11}, p_{11}, p_{10}, p_{12}, f_{13}, p_{17}, p_{16}, p_{12}, p_{15}$ in this order. The first two steps of the derivation are shown below: $\begin{matrix} f_1 & \\ d_2 & d_2 \end{matrix} \Longrightarrow_{p_1} \begin{matrix} & p & \\ f_1 & d & d \\ d_2 & d_2 & \end{matrix} \Longrightarrow_{p_2}$

$$\begin{matrix} & & d & & \\ & p & p & p & \\ f_1 & d & d & & \\ d_2 & d_2 & & & \end{matrix}$$

It can be seen that the contextual array grammar rules of G generate the picture language L_1. In an array M of L_1 if we replace each of the symbols by the corresponding primitive "kolam" pattern as in Fig. 2, then we obtain the "kolam" itself. For example the rectangular array in Fig. 1(B) yields the "kolam" in Fig. 3, when the symbols are substituted in this manner.

Informally expressed, the basic model of a rewriting array P system has a membrane structure μ of a P system with m membranes, denoted by a well-formed expression of parentheses over the alphabet of left and right parentheses $[_i$ and $]_i$, $1 \leq i \leq m$. For example, $[_1\,[_2\,]_2\,[_3\,]_3\,]_1$ means that membrane 1 is the outermost membrane (also called skin membrane), which contains membranes 2 and 3. The membranes (also called regions) have arrays as objects and rewriting rules as evolution rules, which can be applied to the objects. The application of a rule at the level of a membrane is sequential but the objects in all the membranes are rewritten in a maximal parallel way in the sense that all objects that can be rewritten in the membranes should be rewritten. There is a target indication *here*, *in* or *out* associated with every rule. The evolved array in a membrane is retained in the same membrane if the target is *here* and sent to an immediately inner membrane or outer membrane according as the target is *in* or *out*.

We now recall contextual array P system that has in its membranes, picture array objects and contextual array rules as in a contextual array grammar [4].

Definition 1 [3]. *A contextual array P system with $m \geq 1$ membranes is a construct $\Pi = (V, \#, \mu, X_1, \ldots, X_m, R_1, \ldots, R_m, i_o)$, where V is the alphabet; $\#$ is the blank symbol; μ is a membrane structure with m membranes or regions, labeled by $1, \ldots, m$, in a one-to-one manner; $X_1, \ldots, X_m$ are finite sets of arrays over V associated with the m regions of μ; $R_1, \ldots, R_m$ are finite sets of contextual array rules over V associated with the m regions of μ; the rules have attached targets here, out, in, or in_j, $1 \leq j \leq m$; i_o is the label of a membrane called output membrane, wherein the results of successful computations are collected.*

A *computation step* in a contextual array P system is done as follows: for each array $\mathcal{A}$ in each region of the system, if a contextual array production p in the region can be applied to $\mathcal{A}$, then it should be applied which means that the application of a rule is sequential at the level of arrays, but maximally parallel at the level of the whole system. If more than one rule is applicable at the same time, then one is chosen in a nondeterministic way. The resulting array, if any, is placed in the region indicated by the target associated with the rule having been applied with interpreting the attached target as follows: *here* means that the array remains in the same region, *out* means that the array exits the current membrane and is placed in the immediately outer membrane if one exists (we do not allow the target *out* to be used by a rule assigned to the skin membrane), in_j means that the array is immediately sent to the directly inner membrane with label j, and *in* means that the array is immediately sent to one of the directly inner membranes, chosen in a nondeterministic way if several such membranes exist (if no inner membrane exists, then a rule with the target indication *in* cannot be used). A computation is called *successful* if and only if it halts, which means that a configuration has been reached where no rule can be applied to the existing arrays. The result of a halting computation consists of the arrays collected in the membrane with label i_o in the halting configuration.

A variant of contextual array P system is considered in [10]. In this variant, instead of a sequential application of a contextual array grammar rule, a set of

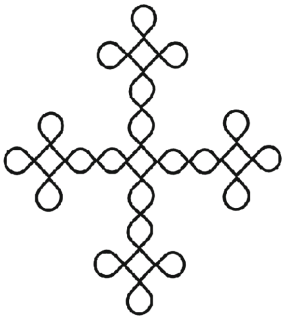


Fig. 4. P system "kolam".

$$
\begin{array}{ccccccccc}
 & & & d_1 & & & & & \\
 & & d_3 & d & d_4 & & & & \\
 & & & p_v & & & & & \\
 & d_1 & & p_v & & & d_1 & & \\
d_3 & d & p_h & p_h & d & p_h & p_h & d & d_4 \\
 & d_2 & & p_v & & & d_2 & & \\
 & & & p_v & & & & & \\
 & & d_3 & d & d_4 & & & & \\
 & & & d_2 & & & & &
\end{array}
$$

Fig. 5. Array representing P system "kolam".

rules in a membrane with the same target is applied in parallel. The resulting array P system is called Parallel contextual array P system ($PCAP$). It has been shown in [10], that the advantage of this kind of a $PCAP$ system is that the number of membranes required in the construction of such systems in generating picture languages, is reduced in comparison with the sequential CAP.

3.1 "Kolam" Generation Using $PCAP$

We now make use of $PCAP$ in generating "Kolam" patterns. We consider the kind of "kolam" shown in Fig. 4. This kind of a kolam can be represented by an array as shown in Fig. 5. We consider the picture language L_2 whose elements are picture arrays, one member of which is shown in Fig. 5. In fact in an array in L_2, the middle vertical column is a word of the form $d_1(dp_vp_v)^{n-1}d(p_vp_vd)^{n-1}d_2$, $n \geq 1$ but written vertically while every symbol d in this column except the central symbol d has the symbol d_3 to its immediate left and the symbol d_4 to its immediate right. Also, the middle horizontal row is a word of the form $d_3(dp_hp_h)^{n-1}d(p_hp_hd)^{n-1}d_4$, $n \geq 1$ while every symbol d in this row except the central symbol d has the symbol d_1 immediately above it and the symbol d_2 immediately below it. The picture language L_2 is generated by a $PCAP$ Π with a membrane structure $[_1\,[_2\,]_2\,]_1$ as in Fig. 6. The region labelled 1 has an axiom array d while the region 2 has no array inside it initially. The region 2 is the output membrane where the picture arrays generated are collected which constitute the arrays in L_2. Region 1 has four rules r_1, r_2, r_3, r_4

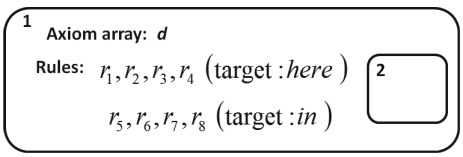


Fig. 6. Membrane structure

with target *here* and another four rules r_5, r_6, r_7, r_8 with target *in* while region 2 has no rules. The rules are given below:

$$r_1 := \begin{matrix} & d_3 & d & d_4 & \\ & & p_v & & \\ & & p_v & & \\ & & \boxed{d} & & \end{matrix}, \quad r_2 := \begin{matrix} & & d_1 & & \\ d & p_h & p_h & \boxed{d} & \\ & & d_2 & & \end{matrix}, \quad r_3 := \begin{matrix} & & \boxed{d} & & \\ & & p_v & & \\ & & p_v & & \\ d_3 & d & d_4 & & \end{matrix}, \quad r_4 := \begin{matrix} & & d_1 \\ \boxed{d} & p_h & p_h & d \\ & & d_2 \end{matrix}$$

$$r_5 := \begin{matrix} d_1 \\ \boxed{d} \end{matrix}, \quad r_6 := d_3 \boxed{d}, \quad r_7 := \begin{matrix} \boxed{d} \\ d_2 \end{matrix}, \quad r_8 := \boxed{d}\, d_4$$

The generation of an array in L_2 as in Fig. 5 in the $PCAP$ Π takes place as follows: Starting with the axiom array d in region 1, the rules r_1, r_2, r_3, r_4 are applied in parallel generating the array

$$\begin{matrix} & & d_3 & d & d_4 & & \\ & & & p_v & & & \\ d_1 & & & p_v & & & d_1 \\ d & p_h & p_h & d & p_h & p_h & d \\ d_2 & & & p_v & & & d_2 \\ & & & p_v & & & \\ & & d_3 & d & d_4 & & \end{matrix}.$$

The array remains in region 1 as the target indication of the rules r_1, r_2, r_3, r_4 is *here*. The process can be repeated. When the rules r_5, r_6, r_7, r_8 are applied in parallel, then the array generated is sent to region 2, due to the target *in* in the rules r_5, r_6, r_7, r_8. Since region 2 is the output membrane and does not contain any rule, the array is collected in the picture language L_2.

In every array in L_2 if we replace as in Narasimhan's method [8], the label (or symbol) of every pixel by the corresponding primitive "kolam" pattern from the list in Fig. 2, then we obtain a "kolam" which will be an enlarged version of the "kolam" in Fig. 4. In fact when the rules r_1, r_2, r_3, r_4 are applied in parallel once and the rules r_5, r_6, r_7, r_8 are then applied once in parallel, the array generated is as in Fig. 5. The "kolam" derived from this array by Narasimhan's method is indeed the "kolam" shown in Fig. 4.

We have shown that a class of "kolams", an example of which is shown in Fig. 4, can be generated by the P system, namely, parallel contextual array P system ($PCAP$) [10]. It can be shown that the same "kolam" class can be generated by a contextual array P system (CAP) [3] but it will require more number of membranes while the $PCAP$ considered requires only two membranes and thus is a simpler model for "kolam" generation. But a $PCAP$ with one membrane is not enough to generate this "kolam" class as the rules can have the only target *here* and so the rules r_1 to r_8 need not be applied in the combination of rules mentioned earlier, namely, r_1, r_2, r_3, r_4 together and r_5, r_6, r_7, r_8 together in parallel. But then the number of pixels in the vertical middle column in Fig. 4 above and below the central symbol d as well as in the horizontal middle row in Fig. 4, in the right and left of the central d, need not be the same.

4 Concluding Remarks

We have considered the problem of generation of kolam-designs, in the framework of the computability model of P system. The intricate kolam designs are very good examples of visual expressions of creative thought [1] and can be considered as diagrams that could be classified under "art" in a very broad sense. The contextual array P (CAP) system model considered here makes use of certain kinds of rules of object generation, thus providing a novel approach to handle such kolam designs. It will be of interest to consider more complex "kolam" patterns and examine the ability of the contextual array P system in handling such patterns.

Acknowledgement. The authors thank the reviewers for their constructive comments and suggestions which very much helped to revise the paper and present the details in a better form. The first author Ibrahim Venkat acknowledges the support for this research by the RUI Grant Account # 1001/PKOMP/811290 awarded by Universiti Sains Malaysia. The third author K.G. Subramanian is grateful to University Grants Commission, India, for the award of Emeritus Fellowship (No. F.6-6/2016-17/EMERITUS-2015-17-GEN-5933 / (SA-II)) to him to execute his work in the Department of Mathematics, Madras Christian College.

References

1. Ascher, M.: The kolam tradition. Am. Sci. **90**, 56–61 (2002)
2. Ceterchi, R., Mutyam, M., Paun, G., Subramanian, K.G.: Array-rewriting P systems. Nat. Comput. **2**(3), 229–249 (2003)
3. Fernau, H., Freund, R., Schmid, M.L., Subramanian, K.G., Wiederhold, P.: Contextual array grammars and array P systems. Ann. Math. Artif. Intell. **75**(1–2), 5–26 (2015)
4. Freund, R., Paun, G., Rozenberg, G.: Contextual array grammars. In: Subramanian, K.G., et al. (eds.) Formal Models, Languages and Applications, pp. 112–136. World Scientific Publishing, Singapore (2007)
5. Nagata, S., Robinson, T.: Digitalization of kolam patterns and tactile kolam tools. In: Subramanian, K.G., et al. (eds.) Formal Models, Languages and Applications, pp. 354–363. World Scientific Publishing, Singapore (2007)
6. Păun, G.: Membrane Computing: An Introduction. Springer, Heidelbrg (2000). https://doi.org/10.1007/978-3-642-56196-2
7. Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages, vol. 3. Springer, Berlin (1997). https://doi.org/10.1007/978-3-642-59126-6
8. Siromoney, G., Siromoney, R., Krithivasan, K.: Array grammars and kolam. Comput. Graph. Image Process. **3**(1), 63–82 (1974)
9. Subramanian, K.G., Saravanan, R., Robinson, T.: P systems for array generation and application to kolam patterns. Forma **22**, 47–54 (2007)
10. Subramanian, K.G., Bera, S., Song, B., Pan, L., Zhang, Z.: Array P systems based on parallel rewriting with array contextual rules. In: Pre-Proceedings of Asian Conference on Membrane Computing (ACMC 2017), pp. 403–415 (2017)