International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 8 (2016) pp. 326–335

Hybrid P System with Conditional Communication

M. Nithya Kalyani[1], **P. Helen Chandra** [2] **and S. M. Saroja T. Kalavathy**[3]

[1]Research Scholar in Mathematics,
Jayaraj Annapackiam College for Women (Autonomous),
Periyakulam, Theni District, Tamilnadu, India
rnithraj@gmail.com

[2,3] Associate Professor in Mathematics,
Jayaraj Annapackiam College for Women (Autonomous),
Periyakulam, Theni District, Tamilnadu, India
[2]*chandrajac@yahoo.com*, [3] *kalaoliver@gmail.com*

Abstract: **The feature of conditional communication in membrane computing has been introduced in symbol objects and arrays in [1, 8, 9, 19]. Hybridity and context free is a special feature which has been applied on Hybrid P system [17]. In this paper we study the nature of membrane computing with conditional communication in Hybrid context free puzzle P system and examine the power of the system by comparing the model with certain array grammars generating languages.**

Keywords: Context-free puzzle grammar, P system, Array Rewriting P system, Hybrid prescribed team, Hybrid P system, Conditional communication.

I. Introduction

The study of two-dimensional grammar models is an area of investigation motivated by different problems in the frame work of image analysis and picture processing [6]. Motivated by problems of tiling in the two-dimensional plane, one such syntactic method called puzzle grammar system was proposed by Nivat et al. and investigated in [12] for its properties by comparing with different array grammars. A subclass of puzzle grammar called context-free puzzle grammars with rules of a specific nature was introduced by Subramanian et al. [18]. In the area of grammar system, Dassow et al. [4] have introduced cooperating array grammar system extending the notion of cooperating distributed (string) grammar system to arrays. The notion of a team CD grammar system was introduced and investigated by removing the restriction that at each moment only one component is enabled [3, 10, 11, 13]. Fernau [5] and Maurice ter Beek [11] studied hybrid (prescribed) team CD grammar system allowing work to be done in teams while at the same time assuming these teams to have different capabilities.

On the other hand, research on membrane computing was initiated by Paun [14] introducing a new computability model called P system, which is a distributed, highly parallel theoretical computing model based on the membrane structure and the behavior of the living cells. Among a variety of applications of this model, the problem of handling array languages using P system has been considered by Ceterchi et al. introducing array rewriting P system [2] and thus linking the two areas of membrane computing and picture grammars. A kind of array P system with objects in the regions as arrays and the productions as hybrid prescribed team of CD grammar rules was introduced in [7], which allow work to be done in team with the possibility of different teams having different modes of derivation. A Hybrid P system was introduced in [17] considering context-free puzzle grammar rules instead of context-free or regular array rewriting rules. Different classes of P systems with conditional communication have been introduced and studied for its computational power [1, 8, 9 ,19].

In this paper a new computing model called Hybrid P system with conditional communication is introduced by considering context-free puzzle grammar rules. Comparison is done with the parallel array rewriting P system [21] and CD grammar system [4]. As an application of our HP system, we have generated certain *floor designs*.

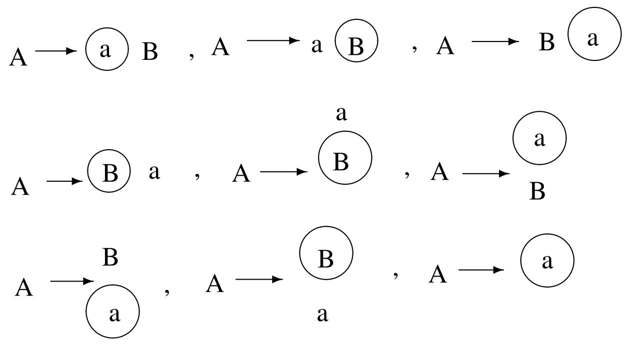
II. Preliminaries

In this section, we recall some prerequisites necessary for understanding the sequel.

A. *Context-Free Puzzle Grammar* $(CFPG)$ *[18]*

A basic puzzle grammar (BPG) is a structure $G = (N, T, P, S)$ where N and T are finite sets of symbols; $N \cap T = \phi$. Elements of N are called non-terminals and elements of T, terminals. The start symbol or the axiom is $S \in N$. The set P consists of rules of the forms as in $Fig.A$.

A context-free puzzle grammar $(CFPG)$ is a structure $G = (N, T, P, S)$ where N, T, S are as above and P the set of rules of the form $A \rightarrow \alpha$ where α is a finite, connected array of one or more cells, each cell containing a nonterminal or a terminal symbol, with a symbol in one of the cells of α being circled. Derivations are done as in BPG.



where $A, B \in N$ and $a \in T$.

Fig.A BPG Rules

B. Hybrid Prescribed Team Context-free puzzle grammar System [17]

A hybrid prescribed team CD grammar system [11] is a construct
$\Gamma = (N, T, P_1, \ldots, P_n, S, (Q_1, f_1), (Q_2, f_2), \ldots, (Q_m, f_m))$
where $N, T, P_1, \ldots, P_n$ are defined as in the cooperating array grammar system [4]. $Q_1, Q_2, \ldots Q_m$ are teams over $N \cup T$, multiset of sets of productions $P_1, \ldots, P_n$ and $f_1, f_2, \ldots, f_m$ are modes of derivation.

For a team $Q_i, 1 \leq i \leq m, Q_i = \{P_{ij} | 1 \leq j \leq m_i\}$, and two arrays D_1 and $D_2 \in (N \cup T)^+$ a direct derivation step is defined by $D_1 \vdash_{Q_i} D_2$ if and only if there are array productions $p_j \in P_{ij}, 1 \leq j \leq m_i$, such that in D_1 we can find m_k non-overlapping areas such that the sub-patterns of D_1 located at these areas coincide with the left-hand sides of the array productions p_j and yield D_2 by replacing them by the right-hand sides of the array productions p_j.

An application of the team Q_i to an array D_1 therefore means the following: from each set P_{ij}, one array production p_j is chosen such that $P_1, \ldots, P_m$ can be applied in a parallel manner to D_1 without disturbing each other. Note that the array productions p_j need not all be different although coming from different sets within the team Q_i. The derivation relations are defined by $\vdash^{*}_{Q_i}$, $\vdash^{=k}_{Q_i}$, $\vdash^{<k}_{Q_i}$, $\vdash^{>k}_{Q_i}$ and $\vdash^{t}_{Q_i}$ respectively. i.e, derivations i^{th} team Q_i of arbitrary, of exactly k successive steps, of at most k steps, of atleast k steps and of as many steps as possible, respectively; this maximal derivation mode t is defined more precisely by: $D_1 \vdash^{t}_{Q_i} D_2$ if and only if $D_1 \vdash^{*}_{Q_i} D_2$ and there is at least one component P_{i,j_0} in the team Q_i such that no array production in P_{i,j_0} can be applied to D_2 anymore. Note that in the t-mode a derivation with a team Q_i can be blocked, although in every $P_{i,j}$ we can find an array production which is applicable to the underlying array.

The language generated by Γ is

$$L(\Gamma) = \left\{ X \in T^{**}/S \underset{Q_1}{\overset{f_1}{\Rightarrow}} X_1 \underset{Q_2}{\overset{f_2}{\Rightarrow}} X_2 \Rightarrow \ldots \Rightarrow \underset{Q_m}{\overset{f_m}{\Rightarrow}} X_m = X \right\}$$

A Hybrid context-free puzzle grammar system with prescribed teams $(PTHCFPGS)$[17] is a construct
$\Gamma = (N, T, P_1, \ldots, P_n, S, (Q_1, f_1), (Q_2, f_2), \ldots, (Q_m, f_m))$
where N, T, S and (Q_i, f_i), $i = 1, 2, \ldots, m$ are defined as in the Hybrid prescribed team CD grammar system and $P_i, i = 1, 2, ..., n$ are non-empty finite sets of context-free puzzle grammar rules over $N \cup T$.

For a Hybrid Context-free puzzle grammar system with prescribed teams Γ, the array language generated by Γ is

$$L(\Gamma) = \left\{ X \in T^{**}/S \underset{Q_1}{\overset{f_1}{\Rightarrow}} X_1 \underset{Q_2}{\overset{f_2}{\Rightarrow}} X_2 \Rightarrow \ldots \Rightarrow \underset{Q_m}{\overset{f_m}{\Rightarrow}} X_m = X, m \geq 1 \right\}$$

The family of array languages generated by a $PTHCFPGS$ with at most n components is denoted by $PTH_n(CFPGL), n \geq 1$.

C. Array-Rewriting P system [2]

The array-rewriting P system (of degree $m \geq 1$) is a construct

$$\Pi = (V, T, \#, \mu, F_1, \ldots, F_m, R_1, \ldots, R_m, i_o)$$

where V is the total alphabet, $T \subseteq V$ is the terminal alphabet, # is the blank symbol, μ is a membrane structure with m membranes labeled in a one-to-one way with $1, 2, \cdots, m$, $F_1, \cdots, F_m$ are finite sets of arrays over V associated with the m regions of $\mu, R_1, \ldots, R_m$ are finite sets of array rewriting rules over V associated with the m regions of μ; the rules have attached targets *here, out, in*(in general, *here* is omitted), hence they are of the form $\mathcal{A} \rightarrow \mathcal{B}(tar)$; finally, i_o is the label of an elementary membrane of μ (the output membrane). We emphasize the fact that in an array P system we distinguish terminal and auxiliary symbols in the Lindenmayer sense, that is, no condition is imposed on the symbols appearing in the left hand side of rules. The general case, when a set T is distinguished, we speak about an *extended* P system, when $V = T$ we have a *nonextended* system. According to the form of its rules, an array P system can be monotonic, context-free (CF), #-context-free $(\#CF)$ or regular (REG). In the extended case, a rule is called regular if it is of one of the following forms:

$$a \ \# \rightarrow b \ c \ , \quad \# \ a \rightarrow c \ b \ , \quad \begin{matrix} a \\ \# \end{matrix} \rightarrow \begin{matrix} b \\ c \end{matrix} \ ,$$

$$\begin{matrix} \# \\ a \end{matrix} \rightarrow \begin{matrix} c \\ b \end{matrix} \ , \quad a \rightarrow b$$

where all a, b, c are non-blank symbols. In the non-extended case, we use the notion of a regular rule in the restricted sense; such a rule is of one of the forms:

$$a \ \# \rightarrow a \ b \ , \quad \# \ a \rightarrow b \ a \ ,$$

$$\begin{matrix} \# \\ a \end{matrix} \rightarrow \begin{matrix} b \\ c \end{matrix} \ , \quad \begin{matrix} a \\ \# \end{matrix} \rightarrow \begin{matrix} c \\ b \end{matrix}$$

where all a, b are non-blank symbols.

The set of all arrays generated by a system Π is denoted by $AL(\Pi)$. The family of all array languages $AL(\Pi)$ generated by systems Π as above with at most m membranes and rules of type $\alpha \in \{REG, CF, \#CF\}$ is denoted by $EAP_m(\alpha)$. If non-extended systems are considered, then we write $AP_m(\alpha)$.

D. Array-Rewriting P System with Conditional Communication [8]

A (string) rewriting P system with conditional communication [1] is defined as
$\Pi = (V, T, \#, \mu, M_1, \ldots, M_m,$

$$(R_1, P_1, F_1), (R_2, P_2, F_2), \ldots, (R_m, P_m, F_m))$$

where V is the total alphabet, $T \subseteq V$ is the terminal alphabet, $\#$ is the blank symbol, μ is a membrane structure with m membranes injectively labeled by $1, 2, \cdots, m$. M_i, $1 \leq i \leq m$ denote the finite languages over V representing the strings initially present in the regions $1, 2, \ldots, m$ of the system, $R_i, 1 \leq i \leq m$ are the finite sets of context-free rules over V (without target indications and priority relations) present in the regions $1, 2, \ldots, m$ of the system, P_i and F_i are the permitting and forbidding conditions associated with the regions $i, 1 \leq i \leq m$ which restrict the communication of strings produced in the corresponding regions. The conditions can be of the following forms:

empty: No restriction is imposed on strings, they either exit in the current membrane or enter any of the directly inner membranes freely; we denote an empty permitting condition by $(True, \alpha), \alpha \in \{in, out\}$ and an empty forbidding condition by $(False, not\alpha), \alpha \in \{in, out\}$.

symbol checking: each P_i is a set of pairs (a, α), $\alpha \in \{in, out\}$, for $a \in V$ and each F_i is a set of pairs $(b, not\alpha), \alpha \in \{in, out\}$ for $b \in V$; a string w can go to a lower membrane only if there is a pair $(a, in) \in P_i$ with $a \in alph(w)$ and for each $(b, notin) \in F_i$ we have $b \notin alph(w)$; similarly for the string to go out of membrane i, it is necessary to have $a \in alph(w)$ for at least one pair $(a, out) \in P_i$ and $b \notin alph(w)$ for all $(b, notout) \in F_i$.

substrings checking: each P_i is a set of pairs $(u, \alpha), \alpha \in \{in, out\}$, for $u \in V^+$ and each F_i is a set of pairs $(v, not\alpha), \alpha \in \{in, out\}$ for $v \in V^+$; a string w can go to a lower membrane only if there is a pair $(u, in) \in P_i$ with $u \in Sub(w)$ and for each $(v, notin) \in F_i$ we have $v \notin Sub(w)$; similarly for the string to go out of membrane i, it is necessary to have $u \in Sub(w)$ for at least one pair $(u, out) \in P_i$ and $v \notin Sub(w)$ for all $(v, notout) \in F_i$.

Thus we have conditions of the type empty, symbol, sub_k respectively, where k is the length of the longest string in all P_i, F_i; when no upper bound is imposed we replace the subscript by $*$. A system is said to be non-extended if $V = T$.

The transitions in the system are defined in the following way. In each region, each string which can be rewritten by a rule from that region is rewritten. The rule to be applied and the symbol rewritten by it are non-deterministically chosen. Each string obtained in this way is checked against the conditions P_i, F_i in the respective regions. According to the specified conditions the string will be immediately sent out of the membrane or to an inner membrane if any exists; if it fulfills both in and out conditions, then either it is sent out of the membrane or to an inner membrane (non- deterministically choosing any of the available inner membranes). If a string does not fulfill any condition, or it fulfills only *in* conditions and there is no inner membrane, then the string remains in the same region. A string which is rewritten and a string which is sent to another membrane is *consumed*, no copy of it is available in the next step in the same membrane. If a string cannot be rewritten then it is directly checked against the communication conditions, and as above, it leaves the membrane or remains inside forever depending on the result of this checking. That is rewriting has priority over communication. As usual, a sequence of transitions forms a computation and the result of a halting computation is the set of strings over T sent out of the system during the computation. A computation which never halts gives no output.

The language generated by the above system is denoted by $L(\Pi)$. The family of all languages $L(\Pi)$ generated by the system P_i of degree at most $m \geq 1$ with permitting conditions of type α and forbidding conditions of type β is denoted by $[E]LSP_m(rw, \alpha, \beta)$, $\alpha, \beta \in \{empty, symbol\} \cup \{sub_k, k \geq 2\}$. If the degree of the systems is not bounded, then the subscript m is replaced by $*$.

An extended array-rewriting P system (of degree $m \geq 1$) with conditional communication [8] is a construct
$\Pi = (V, T, \#, \mu, M_1, \ldots, M_m,$

$$(R_1, P_1, F_1), (R_2, P_2, F_2), \ldots, (R_m, P_m, F_m), i_0)$$

where V is the total alphabet, $T \subseteq V$ is the terminal alphabet, $\#$ is the blank symbol, μ is a membrane structure with m membranes injectively labeled by $1, 2, \ldots, m$. $M_i, 1 \leq i \leq m$, denote the finite sets of arrays over V, representing the arrays initially present in the regions $1, 2, \ldots, m$ of the system, $R_i, 1 \leq i \leq m$ are the finite sets of array-rewriting rules over V (without target indications and priority relations) present in the regions $1, 2, \ldots, m$ of the system, P_i and F_i are the permitting and forbidding conditions associated with the regions $i, 1 \leq i \leq m$. The conditions can be in the forms empty, symbols checking or subarray checking, which are defined analogous to the corresponding forms in the string case. The difference is that the objects are arrays. Thus we have the conditions of the types empty, symbols, subarr in all P_i, F_i and i_0 specifies the output membrane.

The transitions in an array-rewriting P system with conditional communication are analogous to the string case. But the result of a halting computation is as defined for array-rewriting P systems.

The set of all arrays computed by an array-rewriting P system Π with conditional communication is denoted by $[E]AL(\Pi)$ with E being omitted when the system is non-extended. The family of array languages generated by systems as above is denoted by $[E]ALP_m(arw, \alpha, \beta), \alpha, \beta \in \{empty, symbol, subarr\}$.

III. Hybrid P System with Conditional Communication

We now introduce a new kind of rewriting P system, called Hybrid P system with conditional communication, in which rewriting of arrays is in team mode and communication is conditional as in [1].

A. Definition

A Hybrid P System of degree $m(m \geq 1)$ with conditional communication is a construct
$\pi = (V, T, \#, \mu, M_1, \ldots, M_m,$

$$(R_1, P_1, F_1), (R_2, P_2, F_2), \ldots, (R_m, P_m, F_m), i_0)$$

where V is the total alphabet, $T \subseteq V$ is the terminal alphabet, $\#$ is the blank symbol, μ is a membrane structure with m membranes labeled in a one-to-one way with $1, 2, \ldots, m; M_1, M_2, \ldots, M_m$ are finite sets of arrays over V initially associated with the m regions of $\mu; R_1, R_2, \ldots, R_m$ are finite sets of prescribed teams of context - free puzzle grammar rules with the derivation modes associated with the m regions of μ, P_i and F_i are the permitting and forbidding conditions associated with the regions $i, 1 \leq i \leq m$. The conditions can be in the forms empty, symbols checking or subarray checking, which are defined analogous to the corresponding forms in $II.D.$ and i_o specifies the output membrane.

A computation in Hybrid P system with conditional communication is defined in the same way as in an array rewriting P system with conditional communication with successful computations being the halting ones; each array, from each region of the system, which can be rewritten is rewritten by a team of rules associated with that membrane, in a specific derivation mode. The array obtained by rewriting is placed in the region indicated by the conditions associated with the rules used. The set of all arrays generated by Hybrid P system with conditional communication is denoted by $HP(\Pi)$.

The family of languages generated by Hybrid P system with conditional communication is denoted by $HP_m(\alpha, \beta)$. Here degree m is the total number of membranes in the whole system, where $\alpha, \beta \in \{empty, symbol, subarr\}$ are the conditional communications.

B. Example

Consider the HP system with conditional communication $HP_3(subarr, subarr)$
$\Pi_1 = (\{S, X, Y, b\}, \{a\}, \#, [_1[_2[_3]_3]_2]_1, S, \phi, \phi,$

$$(R_1, P_1, F_1), (R_2, P_2, F_2), (R_3, P_3, F_3), 3)$$

where $R_1 = \{Q_1, t\}$,

$$P_1 = \left\{ \left(\begin{array}{cc} \# & \# \\ \# & X \end{array}, in \right), \left(\begin{array}{cc} \# & X \\ \# & \# \end{array}, in \right), \left(\begin{array}{cc} \# & \# \\ b & X \end{array}, in \right), \left(\begin{array}{cc} b & X \\ Y & \# \end{array}, in \right) \right\},$$

$$F_1 = \left\{ \left(\begin{array}{cc} \# & Y \\ \# & X \end{array}, notin \right), \left(\begin{array}{cc} \# & X \\ \# & Y \end{array}, notin \right), \left(\begin{array}{cc} Y & \# \\ b & X \end{array}, notin \right), \left(\begin{array}{cc} b & X \\ \# & \# \end{array}, notin \right) \right\},$$

$R_2 = \{Q_2, *\}$,

$$P_2 = \left\{ \left(\begin{array}{cc} X & b \\ \# & \# \end{array}, in \right), \left(\begin{array}{cc} b & b \\ b & \# \end{array}, in \right), \left(\begin{array}{cc} \# & \# \\ b & X \end{array}, in \right), \left(\begin{array}{cc} \# & b \\ \# & Y \end{array}, in \right) \right\},$$

$$F_2 = \left\{ \left(\begin{array}{cc} \# & b \\ \# & \# \end{array}, \alpha \right), \left(\begin{array}{cc} \# & \# \\ \# & b \end{array}, \alpha \right), \left(\begin{array}{cc} \# & \# \\ \# & Y \end{array}, \alpha \right), \left(\begin{array}{cc} \# & \# \\ Y & \# \end{array}, \alpha \right) \right\},$$

$$\alpha \in \{notin, notout\},$$

$R_3 = \{Q_3, t\}$,

$$P_3 = \left\{ \left(\begin{array}{cc} b & b \\ b & \# \end{array}, in \right), \left(\begin{array}{cc} b & b \\ \# & b \end{array}, in \right), \left(\begin{array}{cc} b & \# \\ b & \# \end{array}, in \right), \left(\begin{array}{cc} b & b \\ \# & \# \end{array}, in \right) \right\},$$

$$F_3 = \left\{ \left(\begin{array}{cc} b & b \\ b & b \end{array}, notin \right) \right\},$$

$$Q_1 = \{J_1\}, Q_2 = \{J_2, J_3, J_4\}, Q_3 = \{J_5, J_6\},$$

$$J_1 = \left\{ S \longrightarrow \begin{array}{ccc} X & \textcircled{b} & X \\ & Y & \end{array} \right\}, \quad J_2 = \{ X \longrightarrow \textcircled{b} X \},$$

$$J_3 = \{ X \longrightarrow X \textcircled{b} \}, \quad J_4 = \left\{ Y \longrightarrow \begin{array}{c} \textcircled{b} \\ Y \end{array} \right\},$$

$$J_5 = \{ X \longrightarrow \textcircled{b} \}, \quad J_6 = \{ Y \longrightarrow \textcircled{b} \}.$$

$L(\Pi_1) = L_1$. Initially, the axiom array S is in the skin region and the other regions do not have objects. The rule J_1 in the team Q_1 is applied with the derivation mode t yields $\begin{array}{ccc} X & b & X \\ & Y & \end{array}$. The generated array can leave the membrane, since $\left(\begin{array}{cc} \# & Y \\ \# & X \end{array}, notin \right), \left(\begin{array}{cc} \# & X \\ \# & Y \end{array}, notin \right)$, $\left(\begin{array}{cc} Y & \# \\ b & X \end{array}, notin \right), \left(\begin{array}{cc} b & X \\ \# & \# \end{array}, notin \right)$ do not exist in the array due to the forbidding conditions and it is sent to the region 2. In region 2, the rule R_2 is applied in $*$ mode. After checking the conditions the generated array is sent to the inner region 3. In region 3, the rule R_3 is applied in t mode, then the array of solid rectangle shape is obtained. After checking the conditions the system halts in region 3.

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

$Fig.$ 1 **Token T with arm length 3**

The picture language L_1 consists of token T with all three arms of equal length as in $Fig.$ 1.

C. Theorem

The class of array languages $HP_3(subarr, subarr)$ intersects the class of $HP_4(CFPL)$**[16]**.

This is a consequence of Example $III.B.$ and we note that the language can be generated by the Hybrid P system with context-free puzzle grammar.

D. Theorem

$HP_3(symbol, subarr)$ - $RAL \neq \phi$, where RAL denotes the class of all regular array languages.

Proof

The language L_1 consisting of arrays in the shape of token T with equal arms are generated by Hybrid P system with conditional communication with context-free puzzle grammar rules. But a regular array grammar cannot generate L_1 as the rewriting in a regular array grammar, when it reaches the *junction* in a T shaped array can either proceed horizontally (left or right) or vertically (down) and thus will fail to produce the third arm [15].

E. Theorem

$FIN \subset HP_1(empty, symbol)$ where FIN denotes the finite array languages, which consist of only finite number of arrays.

Proof

If L is a finite array language over V, then $L = A_1, A_2, \ldots, A_n$. This language is generated by the system $HP_1(empty, symbol)$.

For $\Pi_2 = (V, V, \#, M_1, (R_1, P_1, F_1), 1)$, where

$M_1 = L = \{A_1, A_2, \ldots, A_n\}$,
$P_1 = \{True, out\}, F_1 = \{a, notin\}$ for some $a \in A_i, 1 \leq i \leq n$.

Consider the non-extended system,

$$\Pi_2 = (\{a\}, \{a\}, \#, [_1]_1, \left\{ \begin{array}{cc} a & a \\ a & a \end{array} \right\}, (R_1, P_1, F_1), 1)$$

where $R_1 = \{Q_1, *\}, P_1 = \{True, out\}, F_1 = \{a, notin\}$,

$Q_1 = \{J_1, J_2, J_3\}$,

$$J_1 = \{a \rightarrow \begin{array}{cc} a & a \\ \circled{a} & a \end{array}\},\ J_2 = \{a \rightarrow a\ \circled{a}\},$$

$$J_3 = \{a \rightarrow \begin{array}{c} a \\ \circled{a} \end{array}\}.$$

$$\begin{array}{cccc} a & a & a & a \\ a & a & a & a \\ a & a & a & a \\ a & a & a & a \end{array}$$

$Fig.$ 2 **A solid square**

$L(\pi_2) = L_2$. L_2 is the set of all solid squares of $size \geq 2$ as in $Fig.$ 2 .

F. Theorem

$HP_1(symbol, subarr) \subset HP_2(symbol, subarr)$

Proof

The proper inclusion can be seen as follows. The Hybrid P system with conditional communication rules with two membranes generates L shaped array.
$\Pi_3 = (\{S, U', R', d\}, \{d\}, \#, [_1[_2]_2]_1, S, \phi,$

$$(R_1, P_1, F_1), (R_2, P_2, F_2), 2) \quad where$$

$R_1 = \{Q_1, *\}, P_1 = \{(U', R'), in\}$,

$$F_1 = \left\{ \left(\begin{array}{cc} \# & U' \\ \# & R' \end{array}, notin \right), \left(\begin{array}{cc} \# & \# \\ U' & R' \end{array}, notin \right), \left(\begin{array}{cc} \# & U' \\ R' & d \end{array}, notin \right) \right\},$$

$$R_2 = \{Q_2, t\}, P_2 = \phi, F_2 = \left\{ \left(\begin{array}{cc} d & d \\ d & d \end{array}, notin \right) \right\},$$

$Q_1 = \{J_1, J_2, J_3\}, Q_2 = \{J_4, J_5\}$,

$$J_1 = \{S \longrightarrow \begin{array}{cc} U' & \\ \circled{d} & R' \end{array}\},\quad J_2 = \{U' \longrightarrow \begin{array}{c} U' \\ \circled{d} \end{array}\},$$

$$J_3 = \{R' \longrightarrow \circled{d}\ R'\},\quad J_4 = \{U' \longrightarrow \circled{d}\},$$

$$J_5 = \{R' \longrightarrow \circled{d}\}.$$

$L(\Pi_3) = L_3$. The picture language L_3 consists of token L with equal arms as in $Fig.$ 3. But this language L_3 cannot be generated by a $HP_1(symbol, subarr)$ in just a single membrane, as the rules of the Hybrid context-free puzzle grammar cannot maintain equal growth between vertical and horizontal arms.

$$\begin{array}{cccc} d & & & \\ d & & & \\ d & & & \\ d & d & d & d \end{array}$$

$Fig.$ 3 **Token L with equal arms**

G. Theorem

$HP_3(symbol, subarr)$ - $S/PAL(R : R) \neq \phi$, where $S/PAL(R : R)$ denotes the class of all Sequential or Parallel regular array languages.

Proof

Consider the Hybrid context-free P system with conditional communication in the class of $HP_3(symbol, subarr)$.

$$\Pi_4 = (\{A, B, C, x\}, \{x\}, \#, [_1[_2[_3]_3]_2]_1, \left\{ \begin{array}{cc} A & \\ x & C \\ B & \end{array} \right\}, \phi, \phi,$$

$$(R_1, P_1, F_1), (R_2, P_2, F_2), (R_3, P_3, F_3), 3), where$$

$R_1 = \{Q_1, *\}, P_1 = \{(A, B), in\},$

$$F_1 = \left\{ \left(\begin{array}{cc} x & A \\ x & \# \end{array}, notin \right), \left(\begin{array}{cc} A & x \\ \# & x \end{array}, notin \right), \left(\begin{array}{cc} x & B \\ x & \# \end{array}, notin \right), \left(\begin{array}{cc} B & x \\ \# & x \end{array}, notin \right) \right\},$$

$R_2 = \{Q_2, *\}, P_2 = \{(A, B), in\},$

$$F_2 = \left\{ \left(\begin{array}{cc} \# & B \\ \# & x \end{array}, \alpha \right), \left(\begin{array}{cc} B & \# \\ x & \# \end{array}, \alpha \right), \left(\begin{array}{cc} \# & A \\ \# & \# \end{array}, \alpha \right), \left(\begin{array}{cc} A & \# \\ \# & \# \end{array}, \alpha \right) \right\},$$

$$\alpha \in \{notin, notout\},$$

$R_3 = \{Q_3, t\}, P_3 = \{x, in\},$

$$F_3 = \left\{ \left(\begin{array}{cc} x & x \\ x & x \end{array}, \alpha \right), \alpha \in \{notin, notout\} \right\},$$

$Q_1 = \{J_1, J_2\}, Q_2 = \{J_3, J_4\}, Q_3 = \{J_5, J_6\},$

$$J_1 = \{ C \longrightarrow \begin{array}{cc} ⓧ & C \\ A & \end{array} \}, \quad J_2 = \{ C \longrightarrow \begin{array}{cc} & A \\ ⓧ & x \\ & B \end{array} \},$$

$$J_3 = \{ A \longrightarrow ⓧ \}, \quad J_4 = \{ B \longrightarrow \begin{array}{c} ⓧ \\ B \end{array} \},$$

$$J_5 = \{ A \longrightarrow ⓧ \}, \quad J_6 = \{ B \longrightarrow ⓧ \}.$$

Π_4 generates a language L_4 consisting of arrays in the shape of H with the horizontal line in the middle of the vertical ones as in $Fig.$ 4. But a parallel regular array grammar cannot generate L_4 , since in a parallel regular array grammar, a sentential form array contains at most one nonterminal symbol, which means that the number of rows above and below the middle line of x's are not equal[18].

$$\begin{array}{cccc} x & & & x \\ x & & & x \\ x & x & x & x \\ x & & & x \\ x & & & x \end{array}$$

$Fig.$ **4 Array describing the pattern** H

H. Theorem

(i) $CD_3(REG, f)$ - $HP_m(\alpha, \beta) \neq \phi$, for all $m \geq 1$.

(ii) $HP_3(symbol, subarr)$ - $CD_n(REG, f) \neq \phi$, for all $n \geq 1$.

(iii) $HP_m(\alpha, \beta)$ and $CD_n(REG, f)$ are incomparable for $m, n \geq 3$.

Proof

(i) Consider the Cooperating array grammar system [4] in the class of $CD_3(REG, f)$.

$$\Gamma_1 = (\{S, A, A'\}, \{a\}, P_1, P_2, P_3, \#), where$$

$P_1 = \{S\# \rightarrow AS, S\# \rightarrow AA, A' \rightarrow A\},$

$$P_2 = \left\{ \begin{array}{ccc} A & & a \\ & \rightarrow & \\ \# & & A' \end{array} \right\}, P_3 = \{A' \rightarrow a\}$$

$$\begin{array}{ccccc} a & a & a & \ldots & a \\ a & a & a & \ldots & a \\ \vdots & & & \ldots & \\ a & a & a & \ldots & a \\ a & a & a & \ldots & a \end{array}$$

$Fig.$ **5 Rectangle of size** $n \times m$

$L(\Gamma_1)$ is the set of all solid rectangles of size $n \times m$ with $n, m \geq 2$ as in $Fig.$ 5, which is a regular array language generated by $CD_3(REG, f)$. But L does not belong to $HP_m(\alpha, \beta)$ for any $n \geq 1$.

(ii) Consider the Hybrid P system with conditional communication in the class of $HP_3(symbol, subarr)$.

$$\Pi_5 = (\{S, X, Y, Z, e\}, \{e\}, \#, [_1[_2[_3]_3]_2]_1, S, \phi, \phi,$$

$$(R_1, P_1, F_1), (R_2, P_2, F_2), (R_3, P_3, F_3), 3), \quad where$$

$R_1 = \{Q_1, t\}, P_1 = \{(X, Y, Z), in\},$

$$F_1 = \left\{ \left(\begin{array}{cc} e & Z \\ \# & \# \end{array}, notin \right), \left(\begin{array}{cc} X & Z \\ \# & \# \end{array}, notin \right), \left(\begin{array}{cc} Y & Z \\ \# & \# \end{array}, notin \right), \left(\begin{array}{cc} \# & \# \\ X & Y \end{array}, notin \right) \right\},$$

$$R_2 = \{Q_2, *\}, P_2 = \{(X, Y, Z, \alpha), \alpha \in \{in, out\},$$

$$F_2 = \left\{ \left(\begin{array}{cc} e & Z \\ \# & \# \end{array}, \alpha \right), \left(\begin{array}{cc} \# & Z \\ \# & Z \end{array}, \alpha \right), \left(\begin{array}{cc} X & Z \\ \# & \# \end{array}, \alpha \right), \left(\begin{array}{cc} X & Y \\ e & e \end{array}, \alpha \right), \alpha \in \{notin\} \right\},$$

$R_3 = \{Q_3, t\}, P_3 = \{e, in\},$

$$F_3 = \begin{array}{l} \left\{ \left(\begin{array}{cc} e & e \\ e & \# \end{array}, \alpha \right), \left(\begin{array}{cc} e & e \\ \# & e \end{array}, \alpha \right), \right. \\ \left(\begin{array}{cc} e & \# \\ e & e \end{array}, \alpha \right), \left(\begin{array}{cc} \# & e \\ e & e \end{array}, \alpha \right), \end{array}$$

$$\alpha \in \{notin\}\},$$

$J_1 = \{ S \rightarrow \begin{array}{cc} \textcircled{e} & Z \\ X & Y \end{array} \}$, $J_2 = \{ X \rightarrow \begin{array}{c} \textcircled{e} \\ X \end{array} \}$,

$J_3 = \{ Y \rightarrow \begin{array}{cc} \textcircled{e} & Z \\ X & Y \end{array} \}$, $J_4 = \{ Z \rightarrow \textcircled{e}\ Z \}$,

$J_5 = \{ X \rightarrow \textcircled{e} \}$, $J_6 = \{ Y \rightarrow \textcircled{e} \}$,

$J_7 = \{ Z \rightarrow \textcircled{e} \}$.

$$\begin{array}{cccc} e & e & e & e \\ e & e & e & e \\ e & e & e & e \\ e & e & e & e \end{array}$$

Fig. 6 **Solid square**

$L(\Pi_5) = L_5$. L_5 is the set of all solid squares of size $n \geq 2$ over a single symbol $'e'$ as in $Fig.$ 6. Such arrays cannot be generated by a co-operating regular array grammar system of at most n components in the mode f, $f \in F$ [15].

Statement (iii) is a consequence of statement (i) and (ii)

I. Theorem

(i) $HP_4(empty, symbol) \cap CD_2(BPG, t) \neq \phi$.

(ii) The family $HP_3(symbol, subarr)$ contains languages that cannot be described by any $CD_2(BPG, t)$.

Proof

(i) Consider the Hybrid P system with conditional communication

$$\begin{array}{lll} \Pi_6 = & (\{A, B, C, D, A', B', e\}, & \{e\}, \#, \\ & [_1[_2[_3[_4]_4]_3]_2]_1, S, \phi, \phi, \phi, & (R_1, P_1, F_1), \\ & (R_2, P_2, F_2), (R_3, P_3, F_3), & (R_4, P_4, F_4), 4) \end{array}$$

where $R_1 = \{Q_1, t\}, P_1 = \{(A, B), in\},$

$$F_1 = \begin{array}{l} \left\{ \left(\begin{array}{cc} A & A \\ A & \# \end{array}, notin \right), \left(\begin{array}{cc} e & A \\ \# & \# \end{array}, notin \right), \right. \\ \left. \left(\begin{array}{cc} B & e \\ \# & \# \end{array}, notin \right), \left(\begin{array}{cc} \# & B \\ B & B \end{array}, notin \right) \right\}, \end{array}$$

$R_2 = \{Q_2, *\}, P_2 = \{(A', B'), in\},$

$$F_2 = \begin{array}{l} \left\{ \left(\begin{array}{cc} e & \# \\ e & \# \end{array}, \alpha \right), \left(\begin{array}{cc} A' & \# \\ e & \# \end{array}, \alpha \right), \right. \\ \left. \left(\begin{array}{cc} B' & \# \\ e & \# \end{array}, \alpha \right), \left(\begin{array}{cc} \# & e \\ \# & e \end{array}, \alpha \right) \right\}, \end{array}$$

$$\alpha \in \{notin, notout\},$$

$R_3 = \{Q_3, *\}, P_3 = \{(C, D), in\},$

$$F_3 = \begin{array}{l} \left\{ \left(\begin{array}{cc} \# & C \\ e & e \end{array}, \alpha \right), \left(\begin{array}{cc} \# & e \\ C & D \end{array}, \alpha \right), \right. \\ \left. \left(\begin{array}{cc} e & \# \\ C & D \end{array}, \alpha \right), \left(\begin{array}{cc} D & \# \\ e & e \end{array}, \alpha \right) \right\}, \end{array}$$

$$\alpha \in \{notin, notout\},$$

$R_4 = \{Q_4, t\}, P_4 = \{e, in\},$

$$F_4 = \left\{ \left(\begin{array}{cc} e & e \\ e & e \end{array} \right), notin \right\},$$

$Q_1 = \{J_1\}, Q_2 = \{J_2, J_3\}, Q_3 = \{J_4, J_5\},$

$Q_4 = \{J_6, J_7\},$

$J_1 = \{ S \rightarrow A\ \textcircled{e}\ B \}$, $J_2 = \{ A \rightarrow A\ \textcircled{e}, A \rightarrow \begin{array}{c} \textcircled{e} \\ A' \end{array} \}$,

$J_3 = \{ B \rightarrow \textcircled{e}\ B, B \rightarrow \begin{array}{c} \textcircled{e} \\ B' \end{array} \}$,

$J_4 = \{ A' \rightarrow \begin{array}{c} \textcircled{e} \\ A' \end{array}, A' \rightarrow \textcircled{e}\ C \}$,

$J_5 = \{ B' \rightarrow \begin{array}{c} \textcircled{e} \\ B' \end{array}, B' \rightarrow D\ \textcircled{e} \}$,

$J_6 = \{ C \rightarrow \textcircled{e}\ C, C \rightarrow \textcircled{e} \}$,

$J_7 = \{ D \rightarrow D\ \textcircled{e}, D \rightarrow \textcircled{e} \}$.

$L(\Pi_6) = L_6$. The language L_6 consists of arrays of the form in $Fig.$ 7, where the array represents hollow rectangle of $'e'$. This language also can be generated by $CD_2(BPG, t)$.

Consider the Cooperating Basic puzzle grammar system

$$\Gamma_2 = (\{S, A, B, C, D\}, \{e\}, S, P_1, P_2) \quad where$$

$P_1 = \{ S \rightarrow \begin{array}{c} A \\ \textcircled{e} \end{array}, A \rightarrow \begin{array}{c} A \\ \textcircled{e} \end{array}, A \rightarrow \textcircled{e}\ B, B \rightarrow \textcircled{e}\ B,$

$B \rightarrow \textcircled{e}\ C, C \rightarrow \begin{array}{c} \textcircled{e} \\ D \end{array}, D \rightarrow \begin{array}{c} \textcircled{e} \\ D \end{array}, D \rightarrow E\ \textcircled{e} \},$

$P_2 = \{ E \rightarrow \textcircled{e}\ E, E \rightarrow \textcircled{e} \}$.

$$\begin{array}{ccccc} e & e & e & e & e \\ e & & & & e \\ e & e & e & e & e \end{array}$$

Fig. 7 **A hollow rectangle of** $e's$

The language generated consists of array of the form in $Fig.$ 7 where the array represents hollow rectangles of $'e''s$.

(ii) Consider the Cooperating basic puzzle grammar system [15] in the class of $CD_2(BPG, t)$.

$$\Gamma_3 = (\{S, A, B, C, D\}, \{e\}, S, P_1, P_2)$$

where

$P_1 = \{ S \rightarrow \begin{array}{c} A \\ \textcircled{e} \end{array}, A \rightarrow \begin{array}{c} A \\ \textcircled{e} \end{array}, A \rightarrow \textcircled{e}\ B, B \rightarrow \textcircled{e} B,$

$B \rightarrow \begin{array}{c} \textcircled{e} \\ C \end{array}, C \rightarrow \begin{array}{c} \textcircled{e} \\ C \end{array}, C \rightarrow D\ \textcircled{e} \},$

$P_2 = \{ D \rightarrow D\ \textcircled{e}, D \rightarrow \textcircled{e} \}.$

$$\begin{array}{cccc} e & e & e & e \\ e & & & e \\ e & & & e \\ e & e & e & e \end{array}$$

Fig. 8 **hollow square**

The language generated by Γ_3 consists of arrays of the form in $Fig.$ 8 where the array represents hollow square of $'e''s$.

This statement (ii) is a consequence of the fact that, in a $CD_2(BPG, t)$, *growth* in a square array can take place only at the borders as in $Fig.$ 8. But in a $HP_3(symbol, subarr)$ such a growth can take place even in the interior as in $Fig.$ 6.

J. Application to Floor design Pattern Generation [20, 22]

As an application of Hybrid context-free puzzle P system with conditional communication model, we consider the problem of generating the language L_f of picture arrays describing certain *floor designs*.

$$Define \quad \Pi_f = (V, a, \#, [_1[_2[_3]_3]_2]_1, S, \phi, \phi,$$

$$(R_1, P_1, F_1), (R_2, P_2, F_2), (R_3, P_3, F_3), 3)$$

where $V = \{X, Y, Z, W, X', Y', Z', W', c, d\}$,

$R_1 = \{Q_1, t\}, P_1 = \{(X, Y, Z, W), in\}, F_1 = \{S, notin\}$,

$R_2 = \{Q_2, *\}, P_2 = \{(X', Y', Z', W'), in\}$,

$F_2 = \{X, Y, Z, W, notin\}$,

$R_3 = \{Q_3, t\}, P_3 = \{(c, d), in\}$,

$F_3 = \{(X', Y', Z', W'), notin\}$.

$Q_1 = \{J_1\}, Q_2 = \{J_2, J_3\}, Q_3 = \{J_4, J_5\}$,

$J_1 = \{ S \rightarrow \begin{array}{ccc} X & d & Y \\ d & \textcircled{c} & d \\ Z & d & W \end{array} \},$

$J_2 = \{ X \rightarrow \begin{array}{ccc} X' & d & Y \\ d & c & d \\ Z & d & \textcircled{c} \end{array}, Y \rightarrow \begin{array}{ccc} c & d & Y' \\ d & c & d \\ \textcircled{c} & d & c \end{array} \},$

$J_3 = \{ Z \rightarrow \begin{array}{ccc} c & d & \textcircled{c} \\ d & c & d \\ Z' & d & c \end{array}, W \rightarrow \begin{array}{ccc} \textcircled{c} & d & c \\ d & c & d \\ c & d & W' \end{array} \},$

$J_4 = \{ X' \rightarrow \textcircled{c}, Z' \rightarrow \textcircled{c} \},$

$J_5 = \{ Y' \rightarrow \textcircled{c}, W' \rightarrow \textcircled{c} \}.$

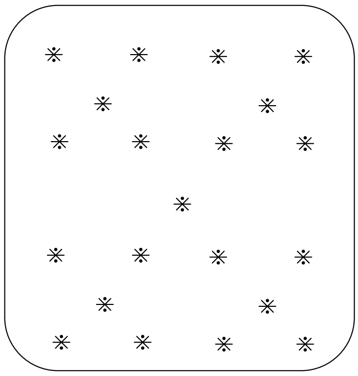


Fig. 9 **The floor design pattern generated by** Π_f

The $HP_3(symbol, subarr)$ system generates the language L_f consisting of arrays over (c, d), with $'c'$ replaced by ⁕ and $'d'$ by a blank square. The language L_f describing *floor design* pattern is shown in $Fig.$ 9.

IV. Conclusion

In this paper, the features of P system with conditional communication are considered with HP system and a new class of P system called Hybrid P system with conditional communication is introduced. Further it is studied for its generating power and used for pattern generation. Also the new system is compared with other pattern generating systems.

References

[1] P. Bottoni, A. Labella, C. Martin-vide, Gh. Paun. Rewriting P systems with conditional communication, *Lecture Notes in computer science*, Springer, Berlin, 2300, pp. 325-353, 2002.

[2] R. Ceterchi, M. Mutyam, Gh.Paun, K.G. Subramanian. *Array Rewriting P Systems, Natural Computing 2*, pp. 220-249, 2003.

[3] E. Csuhaj-Varj, J. Dassow, J. Kelemen, Gh.Paun. Grammar Systems, A grammatical approach to distribution and cooperation, *Topics in Computer Mathematics 5*, Gordon and Breach Science publishers, Y verdon, pp. 233-243, 1994.

[4] J. Dassow, R. Freund, Gh. Paun. Coperating array grammar system, *Int. J. of Pattern Recognition and Artificial Intelligence 9*, pp. 1-25, 1995.

[5] H. Fernau, R. Freund. Bounded Parallelism in Array Grammars used for Character Recognition, Advances in Structural and Syntatical Pattern Recognition, *Proceedings of the SSPR*, Springer, Berlin 1121, pp.40-49, 1996.

[6] D. Giammarresi, A. Restivo, Two dimentional languages, *Handbook of Formal Languages*, Vol. 3, Springer Verlag, pp. 215-267, 1997.

[7] P. Helen Chandra, S. M. Saroja Theerdus Kalavathy. Array P Systems with Hybrid Teams, BIC-TA 2012, *Advances in Intelligent Systems and Computing 201*, Volume 1, Springer, pp. 239-249, 2013.

[8] S. Hemalatha, K. S. Derasanambika, K. G. Subramanian, C. Sri Hari Nagore. Array-Rewriting P systems with conditional communication, *Lecture Note series*, 3, pp. 155-160, 2006.

[9] S. Hemalatha, K. G. Subramanian. Image description based on P Systems with Conditional Communication, *Proceedings of the National Conference on Recent Trends in Mathematical Computing-NCRTMC*, pp. 87-93, 2013.

[10] L. Kari, A. Mateescu, Gh.Paun, A. Salomaa. Teams in cooperating grammar systems, *J. Exper. Th. AI*, 7, pp. 347-359, 1995.

[11] H. Maurice, ter Beek. Teams in grammar systems, Hybridity and weak rewriting, *Acta cybernetica 12*, pp. 427- 444, 1996.

[12] M. Nivat, A. Saoudi, K. G. Subramanian, R. Siromoney, V. R. Dare. Puzzle grammars and Context-free Array Grammars, *Int. J. Pattern Recognition and Artificial Intelligence 5*, pp. 663-676, 1991.

[13] Gh. Paun, G. Rosenberg. Prescribed teams of grammars, *Acta Informatica 31*, pp. 525-537, 1994.

[14] Gh. Paun. Membrane Computing, An introduction, *Springer Verlag*, Berlin, Heidelberg 2002.

[15] R. Saravanan. Cooperation in Basic Puzzle Grammars, *Ph.D. Thesis*, Bharath University, Chennai, India, 2009.

[16] S. M. Saroja Theerdus Kalavathy, P. Helen Chandra. Hybrid Context-free Puzzle Grammars, *Proceedings of the International Conference on Mathematics in Engineering Business Management*, Volume II, pp. 33-38, 2012.

[17] S. M. Saroja Theerdus Kalavathy, P. Helen Chandra, M. Nithya Kalyani. The Power of Hybridity and Context-free in HP System, *Advances in intelligent systems and computing* , Springer, 424, pp. 55-65, 2015.

[18] K. G. Subramanian, R. Siromoney, V. R. Dare, A. Saoudi. Basic Puzzle Languages, *Int. J. Pattern Recognition and Artificial Intelligence 5*, pp. 763-775, 1995.

[19] K. G. Subramanian, S. Hemalatha, C. Sri Hari Nagore, M. Margernstern. On the power of P systems with parallel rewriting and conditional communication, *Romanian Journal of Information Science and Technology*, 10, 2, pp. 137-144, 2007.

[20] K. G. Subramanian, S. Saravanan, T. Robinson. P Systems for array generation and application to Kolam Patterns, *Forma 22*, pp. 47-54, 2007.

[21] K. G. Subramanian, Pradeep Isawasan, Ibrahim Venkat, Linqiang Pan. Parallel Array Rewriting P Systems, *Romanian Journal of Information Science and Technology*, 17, 1, pp. 103-116, 2014.

[22] P. S. Wang. Array Grammars, Patterns and Recognizers, *Series in Computer Science*, World Scientific, 18, pp. 17-41 1989.

Author Biographies

M. Nithya Kalyani received the M.Sc. degree in 2009 and M.Phil. in 2011 from Madurain Kamaraj University, M.Ed. in 2012 from Thiruvalluvar College of Education and M.Phil. (Education) in 2013 from Mother Teresa Womens University, Kodaikanal. She was on the teaching faculty of the Department of Mathematics, Arulmigu Palaniandavar Arts College for Women, Palani from 2013 to 2015. She is doing research in the areas of formal languages and pattern generation.

P. Helen Chandra received the M.Sc. degree from Bharathidasan University, Tiruchi in 1989 , M.Phil. degree from Madurai Kamaraj University, Madurai in 1996 and the Ph.D. degree in Mathematics from the University of Madras, Chennai, India in 2004. Since 1990, she has been the teaching faculty of the department of Mathematics and from 1997 she has been served at different positions like Head of the department of computer science, Vice Principal, Controller of examination and from 2010 onwards serving as the secretary of Jayaraj Annapackiam college for women (Autonomous), Periyakulam, Theni, India. She has presented papers in India and abroad and published papers in national and international journals. Her areas of interest includes formal languages, automata theory, picture languages, DNA computing, membrane computing, artificial cell system and fuzzy mechanism.

S. M. Saroja Theerdus Kalavathy received the M.Sc. degree in 1983 from Jayaraj Annapackiam College for Women(Autonomous), Periyakulam, M.Phil. degree in 1995 from St. Xaviers College, Palayamkottai and the Ph.D. degree in 2013 from Madurai Kamaraj University, Madurai, India. Since 1985, she has been the teaching faculty of the Department of Mathematics, Jayaraj Annapackiam College for Women (Autonomous), Periyakulam, Tamilnadu. Her areas of interest include formal languages, pattern recognition, image processing and other related areas of theoretical computer science.