ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

An Analytic Study on P System with Parallel Basic Puzzle Languages

P. Helen Chandra and S.M. Saroja Theerdus Kalavathy

Associate Professor in Mathematics, Jayaraj Annapackiam College for Women (Autonomous), Periyakulam.

A. Gowthami
Assistant Professor in Mathematics, Nadar Saraswathy College, Theni.

ABSTRACT: A variation of the Basic Puzzle Grammar called Parallel Basic Puzzle Grammar is outlined by incorporating the notion of parallelism in the rewriting of grammar which allows parallel application of the rules to the array rewritten and Array Rewriting P Systems with Parallel Basic Puzzle Grammar $(PBPG)$ is introduced. The power of the rewriting P system with $(PBPG)$ rules is investigated by comparing with array P systems having the rules regular.
.

KEYWORDS: – P system, Basic Puzzle Grammar, Parallel Basic Puzzle Grammar **,** t- communication

I. Introduction

A subclass of puzzle grammars called Basic Puzzle Grammars with rules of a specific nature was introduced by Subramanian et al. [6] and has been studied in [1, 5, 7, 8]. The advantage of the Basic Puzzle Grammar (BPG) is that, it has rules that are "closer" to regular array grammars but is more powerful than regular array grammars in terms of its generating power. Membrane systems are models of computation which are inspired by some basic features of biological membranes. It is a general distributed model, highly parallel, based on the notion of a membrane structure. Such a structure consists of several cell-like membranes, recurrently placed inside a unique skin membrane. For formal definition and generation about P system, we refer to [2, 3, 4]. In this paper, we introduced Array Rewriting P System with Parallel Basic Puzzle Grammar$(PBPG)$. The power of the rewriting P system with $(PBPG)$ rules is investigated by comparing with array P system having the rules regular.

II. PRELIMINARIES

In this section we recall some of the basic definitions related to parallel basic puzzle grammars.

Definition 1

A Basic Puzzle Grammar (BPG) is a structure $G = (N, T, R, S)$ where N and T are finite sets of symbols; $N \cap T = \emptyset$. Elements of N are called non-terminals and elements of T, terminals. $S \in N$ is the start symbol or the axiom. R consists of rules of the following forms:

$$A \to \begin{matrix} a \\ B \end{matrix}, \quad A \to \begin{matrix} a \\ B \end{matrix}, \quad A \to \begin{matrix} B \\ a \end{matrix}, \quad A \to \begin{matrix} R \\ a \end{matrix},$$

$$A \to \begin{matrix} a \\ B \end{matrix}, \quad A \to \begin{matrix} a \\ R \end{matrix}, \quad A \to \begin{matrix} B \\ a \end{matrix}, \quad A \to \begin{matrix} R \\ a \end{matrix}, \quad A \to \begin{matrix} a \\ a \end{matrix}$$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

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

Derivation begins with S written in a unit cell in the two-dimensional plane, with all other cells containing the blank symbol #, not in $N \cup T$. In a derivation step, denoted $\Longrightarrow$, a non-terminal A in a cell is replaced by the right-hand member of a rule whose left-hand side is A. In this replacement, the circled symbol of the right-hand side of the rule used, occupies the cell of the replaced symbol and the non-circled symbol of the right side occupies the cell to the right or the left or above or below the cell of the replaced symbol depending on the type of rule used. The replacement is possible only if the cell to be filled in by the non-circled symbol contains a blank symbol. The set $L(G)$ of picture arrays generated by G, is the set of connected, digitized finite arrays over T, derivable in one or more steps from the axiom.

Definition 2

A Parallel Basic Puzzle Grammar $(PBPG)$ is a structure $G = (V, T, P, A_0)$where V is a finite set of symbols. $V - T$ is the set of non-terminals; $T \subseteq V$ is a finite set of terminal symbols; A_0 is a finite set of axioms which are pictures (finite connected arrays) over V; P is a finite set of tables t_i (i = 1,….,n).

Each table t_i consists of a finite set of rules of the form $X \to M, X \in V; M$ is a picture (or a finite connected array) containing terminals in all cells when $X \in T$, with the terminal in one cell being circled or terminals in all except possibly one cell being circled or terminals in all except possibly one cell in which there is a non-terminal, when $X \in V - T$. Derivations start from an axiom in A_0, with the symbols, for which there are applicable rules in a table t_i, being rewritten in parallel by the rules of t_i; when a rule rewrites a symbol X, the circled symbol in the right side of the rule occupies the cell containing X. When a picture A_1 yields another A_2 by the application of a table t_i then we write $A_1 \Rightarrow_{t_i} A_2$ or $A_1 \Longrightarrow A_2$, when t_i is understood.

The picture language $L(G)$ generated by G consists of all pictures that can be derived from an axiom of A_0 by the tables of rules of G.

Definition 3

A P system of degree m, $(m \geq 1)$, is a construct

$$\Pi = \left(V, T, C, \mu, w_1, \ldots\ldots\ldots, w_m, (R_1, \rho_1), \ldots\ldots\ldots, (R_m, \rho_m), i_0\right)$$

where

i) V is an alphabet; its elements are called objects;

ii) T$\subseteq V$ (the output alphabet);

iii) $C \subseteq V$, $C \cap T = \phi$ (catalysts);

iv) μ is a membrane structure consisting of m membranes, with the membranes and the regions labelled in a one-to-one manner with elements of a given set V; in this section we use the labels 1,2,…, m;

v) $w_i, 1 \leq i \leq m$, are strings representing multi-sets over V associated with the regions 1,2,..., m of μ;

vi) $R_i, 1 \leq i \leq m$, are finite sets of evolution rules over V associated with the regions 1,2,…., m of μ; ρ_i is a partial order relation overR_i, where $1 \leq i \leq m$, specifying a priority relation among rules of R_i. An evolution rule is a pair (u, v), which we will usually write in the form $u \longrightarrow v$, where u is a string over V and $v = v'$or $v = v'\delta$, where v' is a string over

$$\left\{a_{here}, a_{out}, a_{in_j}/a \in V, 1 \leq j \leq m\right\},$$

and δ is a special symbol not in V. The length of u is called the radius of the rule $u \longrightarrow v$. (The strings u, v are understood as representations of multi-sets over V, in the natural sense.)

vii) i_0 is either a number between 1 and m and then it specifies the output membrane of Π, or it is equal to ∞, and then it indicates that the output is read in the outer region.

III. P SYSTEM WITH PARALLEL BASIC PUZZLE GRAMMAR

In this section, we introduce Array Rewriting P System with Parallel Basic Puzzle Grammar$(PBPG)$. The power of the rewriting P system with $(PBPG)$ rules is investigated by comparing with array P system having the rules regular.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

Definition 1

A $PBPG$array P system of degree m (≥ 1) is a construct $\Pi = (V, T, \mu, F_1, F_2, \dots, F_m, R_1, \dots, R_m, i_o)$ where V is the alphabet, $T \subseteq V$ is the terminal alphabet, μ is the membrane structure with m membranes labelled in a one-to-one way with 1,2,…, m, and $F_1, F_2, \dots, F_m$ are finite sets of arrays over V associated with the m regions of $\mu, R_1, \dots, R_m$ are finite sets of $PBPG$ rules containing finite set of tables t_i $(i = 1, \dots, n)$.

Each table t_i consists of a finite set of rules of the form $X \to M$, $X \in V; M$ is a picture (or a finite connected array)containing terminals in all cells when $X \in T$, with the terminal in one cell being circled or terminals in all except possibly one cell being circled or terminals in all except possibly one cell in which there is a non-terminal, when $X \in V - T$.

The rules have attached targets, $here, in, out$ (in general, $here$ is understand and omitted); Finally, i_o is the label of an elementary membrane of μ (the output membrane).

In computation, each array, from each region of the system, which can be rewritten by a rule associated with that region (membrane), should be rewritten in parallel; The array obtained by rewriting is placed in the region indicated by the target associated with the rule used.

The term $here$ means that the array remains in the same region, out means that the array exits the current membrane - 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, i.e., if a computation is reached where no rule can be applied to the existing arrays. The result of a halting computation consists of the arrays composed only of symbols from T placed in the membrane with label i_o in the halting configuration.

The set of all such arrays computed (or generated) by a system Π is denoted by $AL(\Pi)$. The family of all array languages $AL(\Pi)$ generated by system Π as above, with at most m membranes, is denoted by $EAP_m(PBPG)$ if non-extended systems are considered then we write $AP_m(PBPG)$.

Example 2

Consider the array rewriting (Extended $PBPG$) P system,

$$\Pi = \left(\{X, Y, a\}, \{a\}, [_1[_2]_2]_1, \begin{Bmatrix} X & \\ a & Y \end{Bmatrix}, \phi, , R_1, R_2, 2\right)$$

$$R_1 = \left\{ t_1 : X \to \begin{matrix} X \\ a \end{matrix}, Y \to \begin{matrix} a \end{matrix} Y, \ in \right\}, \quad R_2 = \left\{ t_2 : X \to \begin{matrix} a \end{matrix}, \ Y \to \begin{matrix} a \end{matrix}, here \right\}$$

Starting with the axiom array $\begin{Bmatrix} X & \\ a & Y \end{Bmatrix}$ in region 1, the vertical arm and the horizontal arm are grown together, one symbol at a time, by applying in parallel the rules of region 1 . Then the array is sent to region 2 due to target specification in , in the rules of R_1.

In region 2, the non-terminals are changed into the terminal a and the array generated is in the form of the letter L (Figure 1) with equal arms of equal length.

$$\begin{matrix} a \\ a \\ a \\ a \quad a \quad a \quad a \end{matrix}$$

Figure 1 L shaped angle with equal arms

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

Example 3

Consider the array rewriting (Extended) P system,

$$\Pi = \left(\{A,B,C,a\},\{a\},[_1[_2[_3]_3]_2]_1,\begin{Bmatrix} A & a & B \\ & C & \end{Bmatrix},\phi,\phi,R_1,R_2,R_3,3\right)$$

$$R_1 = \left\{t_1: A \to A\overset{a}{\bigcirc},B \to a\overset{B}{\bigcirc},C \to \overset{a}{\underset{C}{\bigcirc}},in\right\}$$

$$R_2 = \left\{t_2: A \to \overset{a}{\bigcirc},B \to \overset{a}{\bigcirc},C \to \overset{a}{\bigcirc},in\right\},\quad R_3 = \phi.$$

Starting with the array $\begin{Bmatrix} A & a & B \\ & C & \end{Bmatrix}$ initially present in region 1 of the system the rules in region 1, can be used in parallel. Then the "horizontal" and "vertical" arms are grown together, one symbol at a time. If the target indication here is used, the same process will be continued in the same membrane. If the target indication *in* is used the array enters the region 2, where in all the three rules are used in parallel and due to the target indication in , the array enters the region 3. Then the array generated is in the form of the letter T (Figure 2) with arms of equal length.

$$\begin{matrix} a & a & a & a & a \\ & & a & & \\ & & a & & \end{matrix}$$

Figure 2 *T*-shaped angle with equal arm.

Theorem 4

$$EAP_2(PBPG) \cap AP_3(REG) \neq \phi$$

Proof

Consider the array rewriting (Extended $PBPG$) P system,

$\Pi = \left(\{X,Y,a\},\{a\},[_1[_2]_2]_1,\begin{Bmatrix} X & \\ a & Y \end{Bmatrix},\phi,,R_1,R_2,2\right)$

$$R_1 = \left\{t_1: X \to \overset{X}{\underset{a}{\bigcirc}},Y \to \overset{a}{\bigcirc}Y,in\right\}$$

$$R_2 = \left\{t_2: X \to \overset{a}{\bigcirc}Y \to \overset{a}{\bigcirc},here\right\}$$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

The language L consists of arrays in the shape of token L with equal arms is generated by the parallel BPG P system (Example 2). This language is also generated by $AP_3(REG)$ with at most 3 membranes with rules regular.

Consider first the array-rewriting (non-extended context-free) P system

$$\Pi = \left(\{a\}, \{a\}\ \#, [_1[_2[_3]_3]_2]_1, \begin{Bmatrix} a \\ a \end{Bmatrix}, \phi, \phi, R_1, R_2, R_3, 3\right)$$

$$R_1 = \left\{ \begin{matrix} & \# \\ \# & a \end{matrix} \rightarrow \begin{matrix} & a \\ \# & a \end{matrix} \ (in) \right\},$$

$$R_2 = \left\{ \begin{matrix} a & \# \\ \# & \end{matrix} \rightarrow \begin{matrix} a & a \\ \# & \end{matrix} \ (out), \begin{matrix} a & \# & \# \\ \# & & \end{matrix} \rightarrow \begin{matrix} a & a & a \\ \# & & \end{matrix} \ (in) \right\},$$

$R_3 = \phi.$

Starting from the unique array $\begin{matrix} a \\ a \end{matrix}$ present initially in region 1, one pixel is grown up in region 1 (which is the skin membrane) in the "vertical arm of an L- shaped angle" (to be formed) and the array is sent to the region 2 due to target specification in in the rule in region 1. In the region 2, one pixel is grown to the right in the "horizontal arm of the L- shaped angle" if the first rule with target indication out is applied and the array is sent back to membrane 1 and the process repeats; instead in region 2 at any moment, if the second rule with target indication in is applied two pixels are added to the horizontal arm and the L- shaped angle formed is sent to membrane 3 and the computation stops. Thus $AL(\Pi)$ consists of all L- shaped angles with equal arms. (Figure 1)
This proves $EAP_2(PBPG) \cap AP_3(REG) \neq \phi$.

Theorem 5

Solid regular hexagons $(H_n : n \geq 2)$ can be generated by a Parallel Basic Puzzle Grammar array rewriting P system.

Proof

Consider the array rewriting (Extended $PBPG$) P system,

$\Pi = (V, T, \mu, A_1, A_2, R_1, R_2, 2)$
where $V = \{A, B, C, E, G, H, x\}$, $T = \{x\}$, $\mu = [_1[_2]_2]_1$
The axiom sets are given by

$$A_1 = \begin{Bmatrix} & A & \\ H & x & B \\ G & x & C \\ & E & \end{Bmatrix}, \quad A_2 = \phi$$

The set of rules are given by,

$$R_1 = \left\{ t_1 : A \rightarrow \begin{matrix} & A & \\ H & & B \\ & x & \end{matrix}, \ B \rightarrow \begin{matrix} & \\ x & B \end{matrix}, \ H \rightarrow H \begin{matrix} \\ x \end{matrix} \ (here, in) \right\}$$

$$R_2 = \{ t_2 : A \rightarrow \begin{matrix} \\ x \end{matrix}, \ B \rightarrow \begin{matrix} \\ x \end{matrix}, \ H \rightarrow \begin{matrix} \\ x \end{matrix}, \ C \rightarrow \begin{matrix} \\ x \end{matrix}, \ G \rightarrow \begin{matrix} \\ x \end{matrix}, \ E \rightarrow \begin{matrix} \\ x \end{matrix} \ \}$$

Starting with the array A_1 initially present in region 1 of the system the rules in region 1 can be used in parallel as they have the same target indication, the parallel application of these rules grows the non-terminals H, A and B together upwards and the process repeats by using the target indication $here$.
If the target indication in is used, the array enters into the region 2, where the rules are applied in parallel and the non-terminals are changed into terminals. Then the array generated is in the form of a solid regular hexagon (Figure 3)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

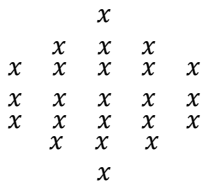

Figure 3 A Solid Regular Hexagon

Definition 6

A P system (of degree $m \geq 1$) with array objects and t −communication (or in short, a t −communicating array P system of type α, $(tEAP_m(\alpha, PBPG))$, $\alpha \in \{t_{\text{in}}, t_{\text{out}}\}$, is a construct

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

Where $V$is the alphabet, $T \subseteq V$ is the terminal alphabet, μ is the membrane structure with m membranes labelled in a one-to-one way with 1,2,…, m, and $F_1, F_2, \ldots, F_m$ are finite sets of arrays over V associated with the m regions of μ, $R_1, \ldots, R_m$ are finite sets of $PBPG$ rules containing finite set of tables t_i $(i = 1, \ldots . n)$ and $tar \in \{in, out\}$; in a given system at most one of the target indications in and out may be present in the rules of the region. i_o is the label of an elementary membrane of μ (the output membrane).

The computation starts in the usual way with the arrays, if any, of F_i initially present in the i^{th} region $1 \leq i \leq m$. Each array in every membrane region i is rewritten by any available array-rewriting rule in the region R_i; only one rule, non-deterministically chosen, is applied to each array in every region. If an array-rewriting rule has target indication in, then the array to which it is applied is non-deterministically sent to one of the immediately direct inner regions and if no such region exists such a rule cannot be applied. If an array-rewriting rule has target indication out, then the array to which it is applied is sent to its immediately direct upper region. If an array-rewriting rule has no target indication, then the array to which it is applied remains in the same region if it can be further rewritten there but if no rule can be applied to it in that region, then one of the following actions is done depending on the system is of type t_{in} or t_{out} .

A t-communicating array P system of type t_{in} has array-rewriting rules in its regions with target indication out or no target indication (and does not have rules with target indication in). In fact when an array cannot be further rewritten in a region, it is sent to the immediately direct inner region if one such region exists. In other words the t −mode or maximal derivation performed enforces the in target command. If the membrane is elementary, the rewritten array remains there.

A t-communicating array P system of type t_{out} has array-rewriting rules in its regions with target indication in or no target indication (and does not have rules with target indication out). In fact when an array cannot be further rewritten in a region, it is sent to the immediately direct outer region if one such region exists. In other words the t −mode or maximal derivation performed enforces the out target command. If the membrane is elementary, the rewritten array remains there.

The set of all arrays computed or generated by a t −communicating P system Π is denoted by $tAL(\Pi)$. The family of all array languages $tAL(\Pi)$ generated by such system Π with at most m membranes and having the $PBPG$ rules is denoted by $tEAP_m(PBPG)$.

Theorem:7

The solid square can be generated by a Parrallel Basic Puzzle Grammar array rewriting P system with t - communication.

Proof:

Consider the t – communication array rewriting P system of type t_{in} $tEAP_3(PBPG)$

$$\Pi = \left(\{C, a\}, \{a\}\ \#, [_1[_2[_3]_3]_2]_1, \begin{Bmatrix} a & c \\ a & a \end{Bmatrix}, \phi, \phi, R_1, R_2, R_3, 3\right)$$

$$R_1 = \left\{t_1 : C \rightarrow \begin{matrix} a & C \\ \sim & a \end{matrix}\right\}$$

$$R_2 = \left\{t_2 : a \rightarrow \begin{matrix} a \\ r \end{matrix}, \ a \rightarrow (x)\, a \ \ (out, in)\right\}$$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

$$R_3 = \{t_3 : C \rightarrow \begin{pmatrix} x \end{pmatrix}\}$$

Starting with the array $\begin{Bmatrix} a & C \\ a & a \end{Bmatrix}$ initially present in region 1 of the system, the rule in region 1 is applied. The non-terminal C is replaced by the array in rule R_1 . Due to the type *tin*, the array enters into region 2 where the rules are applied in parrallel and the array grows horizontally and vertically.
Then the array enters into the region 1 , due to the target indication out and the process repeated. If the target indication in is used, the array enters into region 3, where the non teriminal C is changed into terminal. Then the array generated is in the form of a solid square (Figure 4)

$$\begin{matrix} a & a & a & a & a \\ a & a & a & a & a \\ a & a & a & a & a \\ a & a & a & a & a \\ a & a & a & a & a \end{matrix}$$

Figure 4 A solid square

IV. CONCLUSION AND FUTURE WORK

In this paper, we have introduced a new system by incorporating the parallel version of Basic Puzzle Grammar(BPG) and the array P system with the rules in regions being Parallel Basic Puzzle Grammar rules, resulting a system called P system with Parallel Basic Puzzle Grammar $(PBPG)$.The new model is compared with other generative models. It is worth examining further properties such as closure under set as well as language operations of the system. Comparisons with other such generative models could also be done as future work.

.

REFERENCES

1. P. Laroche, M. Nivat, and A. Saoudi, Context-Sensitivity of Puzzle Grammars, Lecture Notes in Computer Science, Vol.654, pp. 195-212, 1992.
2. Gh. Paun, Computing with cells & atoms, Biddle's Ltd, 2000.
3. Gh. Paun, *Computing with Membranes*, Journal of computer system sciences, 61(1), pp. 108-143, 1998.
4. Gh. Paun, G. Rozenberg, and A. Salomaa, (Eds), The Oxford Handbook of Membrane Computing, Oxford University, 2010.
5. R. Siromoney, A. Huq, M. Chandrasekaran and K.G. Subramanian, Stochastic Puzzle Grammars, *Int. J. of Pattern Recognition and Artificial Intelligence*, **6**, pp. 257-273, 1992.
6. K.G. Subramanian, R. Siromoney, V.R. Dare and A. Saoudi, Basic Puzzle Languages, *Int. J. of Pattern Recognition and Artificial Intelligence*, **5**, pp. 763-775, 1995.
7. K.G. Subramanian, D.G. Thomas, P. Helen Chandra and M. Hoeberechts, Basic Puzzle Grammars and generation of polygons, *Journal of Automata, Languages and Combinatorics*, **6**, pp. 555-568, 2001.
8. K.G. Subramanian, M. Rosihan Ali, K. Atulya Nager, Maurice Margenstern, Array P Systems and t – Communication, Fundamental Informatice , 91, pp. 145-159, 2009.

BIOGRAPHY

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,

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

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 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 area 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. Xavier's College, Palayamkottai and the Ph.D. degree in 2013 from Madurai Kamaraj University, Madurai, Tamilnadu. Since 1985, she has been the teaching faculty of the Department of Mathematics, Jayaraj Annapackiam College for Women (Autonomous), Periyakulam, Tamilnadu. Her area of interest includes formal languages, pattern recognition, image processing and other related areas of theoretical computer science.

A. Gowthami received the M.Sc. and the M.phil. degrees in 2015 and 2016, respectively from Jayaraj Annapackiam College for Women (Autonomous), Periyakulam, Theni, India. At present, she is the teaching faculty of the Department of Mathematics in Nadar Saraswathy College, Theni.