Array P system with t-communicating and permitting mate operation

M. Nithya Kalyani, P. Helén Chandra* and S.M. Saroj́a Theerdús Kalavathy

Department of Mathematics,
Jayaraj Annapackiam College for Women (Autonomous),
Periyakulam, Theni District, Tamilnadu, India
Email: rnithraj@gmail.com
Email: chandrajac@yahoo.com
Email: kalaoliver@gmail.com
*Corresponding author

Abstract: In the field of the DNA computing, splicing system was proposed by Tom Head in which the splicing operation is used. Various models have been studied with mate operation working on strings. The operations mate and drip considered in membrane computing resemble the operations cut and recombination well known from the DNA computing. Recently, a new generative model called array P system with mate operation has been introduced. In this paper, we incorporate the t-communicating and permitting features in the rules of array P system with mate operation and develop the generating picture languages consisting of picture arrays. This enables us to reduce the number of membranes used in the existing array P system models.

Keywords: P system; splicing; mate operation; t-communication; permitting mate operation.

Reference to this paper should be made as follows: Kalyani, M.N., Chandra, P.H. and Kalavathy, S.M.S.T (2017) 'Array P system with t-communicating and permitting mate operation', *Int. J. Artificial Intelligence and Soft Computing*, Vol. 6, No. 3, pp.209–221.

Biographical notes: M. Nithya Kalýani received her MSc in 2009 and MPhil in 2011 from the Madurai Kamaraj University, MEd in 2012 from the Thiruvalluvar College of Education and MPhil (Education) in 2013 from the Mother Teresa Womens University, Kodaikanal. She was a Teaching Faculty of the Department of Mathematics, Arulmigu Palaniandavar Arts College for Women, Palani from 2013 to 2015. She has been researching in the areas of formal languages and pattern generation.

P. Helén Chandra received her MSc from the Bharathidasan University, Tiruchi in 1989, MPhil from the Madurai Kamaraj University, Madurai in 1996 and the PhD in Mathematics from the University of Madras, Chennai, India in 2004. Since 1990, she has been a Teaching Faculty of the Department of Mathematics and from 1997, she has served at the 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 areas of interest includes formal languages, automata theory, picture languages, DNA computing, membrane computing, artificial cell system and fuzzy mechanism.

S.M. Saroj́a Theerdús Kalavathy received her MSc in 1983 from the Jayaraj Annapackiam College for Women(Autonomous), Periyakulam, MPhil in 1995 from the St. Xavier's College, Palayamkottai and the PhD in 2013 from the Madurai Kamaraj University, Madurai, India. Since 1985, she has been a Teaching Faculty of the Department of Mathematics, Jayaraj Annapackiam College for Women (Autonomous), Periyakulam, Tamilnadu. Her areas of interest includes formal languages, pattern recognition, image processing and other related areas of theoretical computer science.

This paper is a revised and expanded version of a paper entitled 'Array P system with permitting mate operation' presented at International Conference on Applicable Mathematics, Stella Maris College (Autonomous), Chennai, 30 November to 1 December 2016.

1 Introduction

DNA computing was introduced more than 20th years ago, when Tom Head formalised the operation of splicing, well known from biology as an operation on DNA strands (Head, 1987). Denninghoff and Pradalier (1989) encoded the range of Turing machines using iterated splicing on multisets. The splicing operation then mainly was used as a basic tool for building a generative mechanism called a splicing system or H system as formalised by Gheorghe Paun.

A new kind of generative mechanism was proposed by Rudolf and Marion (2007) with mate and drip operations working on strings. In P systems and tissue P systems the objects are placed inside the membranes. In the variant of membrane systems introduced by Cardelli (2005), the objects are placed on the membranes. The computations in these models also called *brane calculus* are based on specific ways to divide and fuse membranes and to redistribute the objects on the membranes, the rules usually being applied in a sequential way in contrast to the (maximal) parallel way of applying rules in P systems (Busi, 2005). Various attempts have already been made to combine different models from the area of P systems and of brane calculi (Cardelli and Paun, 2006). Following this research line by investigating tissue P systems with the brane operations mate and drip by Rudolf and Marion (2007), completeness results were obtained both for symbol objects as well as for string objects. It is of interest to note that systems using mate operation are Turing complete and they can compute all Turing computable sets of numbers.

On the other hand, regulating rewriting in a grammar by permitting or forbidding the application of a rule based on the presence or absence of a set of symbols is known in formal language theory (Dassow and Paun, 1989). Picture grammars that use this feature of permitting or forbidding symbols have also been introduced by Ewert and van der Walt (1999).

We incorporate the t-communicating and permitting features in the rules of array P system introduced by Subramanian et al. (2013) with mate operation and develop the generation of picture languages consisting of picture arrays. This enables us to reduce the number of membranes used in the system by comparing to the existing array P system models.

2 Preliminaries

2.1 Array P system with mate operation (Chandra et al., 2016)

An array P system with mate operation is a construct.

$$\Pi = (V, V_T, \#, \mu, A_1, A_2, \ldots, A_m, R_1, R_2, \ldots, R_m, i_0)$$

where V is a finite set of symbols, V_T is a set of terminal symbols, $V_T \subseteq V$, # is the blank symbol, $\# \notin V$, μ is a membrane structure with m membranes injectively labeled by $1, 2, \ldots, m$, $A_1, A_2, \ldots, A_m$ are sequence of sets of axioms where $A_i \subseteq V^{**}$, $1 \leq i \leq m$, describing the initial contents of the membranes, $R_1, R_2, \ldots, R_m$ are finite set of tables containing mate rules associated with the regions of μ, i_0 is the output membrane.

A computation in Π starts with the initial configuration described by A_i. It is performed by applying suitable mate rules from R in a non-deterministic, maximally parallel way, thereby passing from one configuration of the system to the next one. A sequence of transitions constitute a computation.

The mate column operation on a membrane is defined on two arrays X, Y of order $m \times n$ and the rule R_i from the finite set of mate rule $R = (R_1, R_2, \ldots R_m)$ as follows:

$$\{\{(P \mid A, B \mid Q; Z), tar\}, \{\{p \quad \not{c} \quad a \quad \$_c \quad b \quad \not{c} \quad q; z\}, \{ab \to z\}\}\}$$

where $X = S\Phi P\Phi A$ and $Y = B\Phi Q\Phi W; X, Y, Z \subseteq V^{**}$. Mate column operation is done to the elements of arrays X and Y, taking the sub arrays starting from the first row till the end of the row. The sub arrays s, p, a, b, q, w and z of S, P, A, B, Q, W and Z are respectively in the order $p \times q, 1 \leq p \leq m$ and $1 \leq q \leq n$ such that the number of rows are equal and the number of columns may differ. $\{\{p \not{c}\, a\, \$_c\, b \not{c}\, q\,; z\}, \{ab \to z\}\}$ fuses the two arrays carrying $(\,s\,\Phi\,p\,\Phi\,a)$ and $(b\,\Phi\,q\,\Phi\,w)$ into an array which has the form $(\,s\,\Phi\,p\,\Phi\,z\,\Phi\,q\,\Phi\,w)$ where (ab) is replaced by (z). The remaining sub arrays are taken as they are.

The mate row operation on a membrane is defined on two arrays X, Y of order $m \times n$ and the rule R_i from the finite set of mate rule $R = (R_1, R_2, \ldots R_m)$ as follows:

$$\{\{(P \mid A, B \mid Q; Z), tar\}, \{\{p \quad \not{c} \quad a \quad \$_r \quad b \quad \not{c} \quad q; z\}, \{ab \to z\}\}\}$$

where $X = S\Theta P\Theta A$ and $Y = B\Theta Q\Theta W; X, Y, Z \subseteq V^{**}$. Mate row operation is done to the elements of arrays X and Y, taking the sub arrays starting from the first column till the end of the column. The sub arrays s, p, a, b, q, w and z of S, P, A, B, Q, W and Z are respectively in the order $p \times q, 1 \leq p \leq m$ and $1 \leq q \leq n$ such that the number of columns are equal and the number of rows may differ. $\{\{p \not{c}\, a\, \$_r\, b \not{c}\, q\,; z\}, \{ab \to z\}\}$ fuses the two arrays carrying $(\,s\,\Theta\,p\,\Theta\,a)$ and $(b\,\Theta\,q\,\Theta\,w)$ into an array which has the form $(\,s\,\Theta\,p\,\Theta\,z\,\Theta\,q\,\Theta\,w)$ where (ab) is replaced by (z). The remaining sub arrays are taken as they are.

The generated array is sent to the region indicated by tar. If $tar = here$ then the generated array remains in the same membrane where it is generated. If $tar = out$ then the generated array is moved to the region immediately outside the membrane. If $tar = in$ then the generated array is sent to the region immediately inside the membrane.

A computation is successful only if:

1 it halts (which is the case no rule can be applied any more)

2 the output array in the halting configuration is the required array.

The set of all such arrays computed by a system Π is denoted by $MAL(\Pi)$. The family of all array languages generated by systems $MAL(\Pi)$, with at most m membranes with mate operation is denoted by $AP_m(mate)$.

2.2 t-communicating array P system (Subramanian et al., 2009)

A t-communicating array P system of degree $m \geq 1$ and of type t_{in} $(tEAPS_m(t_{in}, CF))$ is a construct.

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

where the components $V, T, \#, \mu, F_1, \dots, F_m, i_o$ are as in an array rewriting P system (Ceterchi et al., 2003) and the rules in the sets $R_1, \dots, R_m$ are CF array rewriting rules of the form $A \to B$. The computation is done in the usual way of starting with the initial arrays (if any) in the regions. The arrays are communicated among the regions in the following manner. If any array-rewriting rule with target indication out, is applied to an array then the resulting array is sent to its immediately direct upper region. If any 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. If no rule can be applied to it in that region, then 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. Note that the system does not have rules with target indication in. The result of a computation is the set of arrays over T collected in the output elementary membrane in the halting configuration.

The family of all array languages generated by a t-communicating array P system of type t_{in}, Π as above, with at most m membranes with rules of type $\alpha \in \{REG, CF\}$ is denoted by $tEAP_m(t_{in}, \alpha)$.

3 Features on mate array P system

We now introduce the notion of a mate array P system with permitting features associated with the mate rules in the regions.

3.1 Permitting mate column/row operation

Permitting mate column/row operation is defined as in array P system with mate operation (Chandra et al., 2016) but the permitting mate column/row rules will be of the form

$$\{\{(P \mid A, B \mid Q; Z), tar\}, \{\{p \quad \not{c} \quad a \quad \$_c \quad b \quad \not{c} \quad q; z\}, \{ab \rightarrow z, per\}\}\},$$

$$\{\{(P \mid A, B \mid Q; Z), tar\}, \{\{p \quad \not{c} \quad a \quad \$_r \quad b \quad \not{c} \quad q; z\}, \{ab \rightarrow z, per\}\}\}$$

provided $per \subseteq \{p, q\}$. If the array satisfies the per condition then the permitting mate column/row operation will be done. If $per = \phi$ then we omit mentioning it in the rule.

3.2 Permitting mate array P system

A permitting mate array P system (of degree $m \geq 1$) is a construct:

$$\Pi = (V, V_T, \#, \mu, A_1, A_2, \ldots, A_m, R_1, R_2, \ldots, R_m, i_0)$$

where

- V is a finite set of symbols
- V_T is a set of terminal symbols, $V_T \subseteq V$
- # is the blank symbol, $\# \notin V$
- μ is a membrane structure with m membranes injectively labeled by $1, 2, \ldots, m$
- $A_1, A_2, \ldots, A_m$ are sequence of sets of axioms where $A_i \subseteq V^{**}$, $1 \leq i \leq m$, describing the initial contents of the membranes.
- $R_1, R_2, \ldots, R_m$ are finite set of tables containing permitting mate rules associated with the regions of μ and $per \subseteq V^{**}$
- i_0 is the output membrane.

A computation in Π starts with the initial configuration described by A_i. It is performed by applying suitable permitting mate rules from R in a non-deterministic, maximally parallel way, thereby passing from one configuration of the system to the next one. A sequence of transitions constitutes a computation. The mate column/row operation is done as in array P system with mate operation but the fusion of two arrays into a single array is done only $per \subseteq \{p, q\}$.

The generated array in the form $S\Phi P\Phi Z\Phi Q\Phi W$ or $S\Theta P\Theta Z\Theta Q\Theta W$ is sent to the region indicated by tar. If $tar = here$ then the generated array remains in the same membrane where it is generated. If $tar = out$ then the generated array is moved to the region immediately outside the membrane. If $tar = in$ then the generated array is sent to the region immediately inside the membrane.

A computation is successful only if:

1 It halts by reaching a configuration where no rule can be applied any more.

2 The output array in the halting configuration is the required array.

The set of all such arrays computed by a system Π is denoted by $pMAL(\Pi)$. The family of all array languages generated by systems $pMAL(\Pi)$, with at most m membranes with mate operation is denoted by $pAP_m(mate)$.

3.3 t-communicating permitting mate array P system

A t-communicating permitting mate array P system of type t_{in} and of degree $m \geq 1$, is the same as a permitting mate array P system of degree $m \geq 1$. Moreover, application of the rules to arrays in the region is done as in a permitting mate array P system and the communication of sending the generated arrays from one region to another is done as in the t-communicating array P system of type t_{in}. As usual, a successful computation is a halting computation with the arrays collected in the output membrane constituting the language generated. The family of picture array languages generated by t-communicating permitting mate array P system of type t_{in} is denoted by $tpAP_m(t_{in}, mate)$.

3.4 Example

Consider the permitting mate array P system.

$$\Pi_1 = \left(\{b, x\}, \{b, x\}, \#, [_1[_2]_2]_1, \left\{ \begin{array}{c} x\,b\,x \\ x\,x\,x \\ x\,b\,x \end{array} \right\}, \phi, (R_1, R_2), 2 \right)$$

where

$$R_1 = \left\{ \begin{array}{l} \left\{ \begin{array}{l|l|l} x\,b\,\ldots\,b & x\;\; x\,b\,\ldots\,b & x\;\; b\,b \\ \vdots\;\vdots\;\ldots\vdots & \vdots\;\;\vdots\;\vdots\;\ldots\vdots & \vdots\;\;\vdots\;\vdots \\ x\,b\,\ldots\,b & x\;\; x\,b\,\ldots\,b & x\;\; b\,b \\ x\,x\,\ldots\,x & x\,,\, x\,x\,\ldots\,x & x\,;\, x\,x\,,(in) \\ x\,b\,\ldots\,b & x\;\; x\,b\,\ldots\,b & x\;\; b\,b \\ \vdots\;\vdots\;\ldots\vdots & \vdots\;\;\vdots\;\vdots\;\ldots\vdots & \vdots\;\;\vdots\;\vdots \\ x\,b\,\ldots\,b & x\;\; x\,b\,\ldots\,b & x\;\; b\,b \end{array} \right\}, \\ \\ \left\{ \begin{array}{l} \big\{ \big\{ b \not{c}\, x\, \$_c\, b \not{c}\, x\,; (b\,b) \big\}, \big\{ (x\,x\,b\ldots b) \to (b\,b), (x\,b, x) \big\} \big\}, \\ \\ \big\{ \big\{ x \not{c}\, x\, \$_c\, x \not{c}\, x\,; (x\,x) \big\}, \big\{ (x\,x\,x\ldots x) \to (x\,x), (x\,x, x) \big\} \big\} \end{array} \right\} \end{array} \right\},$$

$$R_2 = \left\{ \begin{array}{l} \left\{ \begin{array}{lll} x\,b\ldots b\,x & x\,b\ldots b\,x & \\ \vdots\ \vdots\ \vdots\quad \vdots\ \vdots & \vdots\ \vdots\ \vdots\ \vdots\ \vdots & \\ \underline{x\,b\ldots b\,x} & x\,b\ldots b\,x & x\,b\ldots b\,x \\ x\,x\ldots x\,x, & \underline{x\,x\ldots x\,x}; & x\,x\ldots x\,x \\ x\,b\ldots b\,x & x\,b\ldots b\,x & x\,b\ldots b\,x \\ \vdots\ \vdots\ \vdots\quad \vdots\ \vdots & \vdots\ \vdots\ \vdots\ \vdots\ \vdots & \\ x\,b\ldots b\,x & x\,b\ldots b\,x & \end{array} \right\}, \\ \left\{ \begin{array}{l} \left\{ \left\{ x \not c\, x\, \$_r\, x \not c\, x; \begin{pmatrix} x \\ x \\ x \end{pmatrix} \right\}, \left\{ \begin{pmatrix} x \\ x \\ \vdots \\ x \\ \vdots \\ x \\ x \end{pmatrix} \rightarrow \begin{pmatrix} x \\ x \\ x \end{pmatrix}, (x, x) \right\} \right\}, \\ \left\{ \left\{ b \not c\, x\, \$_r\, x \not c\, b; \begin{pmatrix} b \\ x \\ b \end{pmatrix} \right\}, \left\{ \begin{pmatrix} x \\ b \\ \vdots \\ b \\ \vdots \\ b \\ x \end{pmatrix} \rightarrow \begin{pmatrix} b \\ x \\ b \end{pmatrix}, (b, b) \right\} \right\} \end{array} \right\} \end{array} \right\}$$

A computation in Π_1 starts with the initial array $\begin{array}{c} x\,b\,x \\ x\,x\,x \\ x\,b\,x \end{array}$ in region 1, but the region 2 has no initial array. If the permitting mate rule R_1 is applied with the array to itself in region 1, the rules$\{\{b \not c\, x\, \$_c\, b \not c\, x; (b\,b\}, \{(x\,x\,b\ldots b) \rightarrow (b\,b)\}\}$ and $\{\{x \not c\, x\, \$_c\, x \not c\, x; (x\,x\}, \{(x\,x\,x\ldots x) \rightarrow (x\,x)\}\}$ are applicable as the permitting arrays $(xb, x), (xx, x)$ are present. The generated array is moved to region 2, due to the target indication in. In region 2, the permitting mate rule R_2 is applied as the permitting arrays (x, x) and (b, b) are present. The computation is reaching to a halt yielding a H shaped array over $\{x\}$. The picture language generated by Π_1 consists of H shapes with the horizontal line at the middle of the vertical ones as in Figure 1 where 'b' stands for $empty$.

Figure 1 Array describing pattern H

$$\begin{array}{c} \boldsymbol{x}\,b\,b\,b\,\boldsymbol{x} \\ \boldsymbol{x}\,b\,b\,b\,\boldsymbol{x} \\ \boldsymbol{x}\boldsymbol{x}\boldsymbol{x}\boldsymbol{x}\boldsymbol{x} \\ \boldsymbol{x}\,b\,b\,b\,\boldsymbol{x} \\ \boldsymbol{x}\,b\,b\,b\,\boldsymbol{x} \end{array}$$

A vertical bar ‘ | ’ and a horizontal bar ‘ – ’ are used to indicate the place where cutting is done.

3.5 Theorem

A permitting mate array P system generates I shaped arrays with equal arms over {a}.

Proof: Consider the permitting mate array P system

$$\Pi_2 = \left(\{a,x\}, \{a,x\}, \#, [_1[_2]_2]_1, \left\{ \begin{array}{c} a\,a\,a \\ x\,a\,x \\ a\,a\,a \end{array} \right\}, \phi, (R_1, R_2), 2 \right)$$

where

$$R_1 = \left\{ \begin{array}{l} \left\{ \begin{array}{c|c|l} a\ldots a & a\,a\ldots a \quad a\ldots a\,a & a\ldots a \quad a\,a\,a \\ x\ldots x & a\,x\ldots x \quad x\ldots x\,a & x\ldots x \quad x\,a\,x \\ \vdots\ \ \vdots\ \ \vdots & \vdots\ \vdots\ \ \vdots\quad , \vdots\ \ \vdots\ \ \vdots & \vdots\ \ \vdots\ \ \vdots; \vdots\ \vdots\ \vdots, (in) \\ x\ldots x & a\,x\ldots x \quad x\ldots x\,a & x\ldots x \quad x\,a\,x \\ a\ldots a & a\,a\ldots a \quad a\ldots a\,a & a\ldots a \quad a\,a\,a \end{array} \right\}, \\ \\ \left\{ \begin{array}{l} \{\{a \not{c}\, a\,\$_c\, a \not{c}\, a; (a\,a\,a)\}, \{(a\,a\ldots a\ldots a\,a) \rightarrow (a\,a\,a), (a,a)\}\}, \\ \\ \{\{x \not{c}\, a\,\$_c\, a \not{c}\, x; (x\,a\,x)\}, \{(a\,x\ldots x\ldots x\,a) \rightarrow (x\,a\,x), (x,x)\}\} \end{array} \right\} \end{array} \right\},$$

$$R_2 = \left\{ \begin{array}{l} \left\{ \begin{array}{lll} \underline{a\ldots a\,a\ldots a} & \underline{a\ldots a\,a\ldots a} & \\ x\ldots a\,x\ldots x & x\ldots a\,x\ldots x & \\ \vdots\ \vdots\ \ \ \vdots\ \vdots\ \vdots\ \ \ \vdots, & \vdots\ \ \vdots\ \ \vdots\ \vdots\ \ \vdots\ \ \vdots; & \begin{array}{c} x\ldots a\,x\ldots x \\ x\ldots a\,x\ldots x \end{array} \\ x\ldots a\,x\ldots x & x\ldots a\,x\ldots x & \\ a\ldots a\,x\ldots a & a\ldots a\,a\ldots a & \end{array} \right\}, \\ \\ \left\{ \begin{array}{l} \left\{ \left\{ a \not{c}\, x\,\$_r\, a \not{c}\, x; \begin{pmatrix} x \\ x \end{pmatrix} \right\}, \left\{ \begin{pmatrix} x \\ \vdots \\ x \\ a \\ a \end{pmatrix} \rightarrow \begin{pmatrix} x \\ x \end{pmatrix}, \left(a, \begin{pmatrix} x \\ a \end{pmatrix} \right) \right\} \right\}, \\ \\ \left\{ \left\{ a \not{c}\, a\,\$_r\, a \not{c}\, a; \begin{pmatrix} a \\ a \end{pmatrix} \right\}, \left\{ \begin{pmatrix} a \\ \vdots \\ a \\ a \\ a \end{pmatrix} \rightarrow \begin{pmatrix} a \\ a \end{pmatrix}, \left(a, \begin{pmatrix} a \\ a \end{pmatrix} \right) \right\} \right\} \end{array} \right\} \end{array} \right\}$$

The picture language generated by Π_2 consists of I shapes as in Figure 2 where ‘x’ stands for $empty$.

Figure 2 Array describing pattern I

$$\begin{array}{ccccc} \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} \\ x & x & \boldsymbol{a} & x & x \\ x & x & \boldsymbol{a} & x & x \\ x & x & \boldsymbol{a} & x & x \\ \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} \end{array}$$

3.6 Theorem

$$S_s \in tpAP_2(t_{in}, mate).$$

Proof: Consider the t-communicating permitting mate array P system of type t_{in}.

$$\Pi_3 = \left(\{a, x\}, \{a\}, \#, [_1[_2]_2]_1, \left\{ \begin{array}{c} a\,a \\ a\,a \end{array} \right\}, \phi, (R_1, R_2), 2 \right)$$

where

$$R_1 = \left\{ \begin{array}{l} \left\{ \begin{array}{ccc|cccc|cc} a & \dots & a & a & a & \dots & a & a & a \\ \vdots & \vdots & \vdots & \vdots, & \vdots & \vdots & \vdots & \vdots; & \vdots \\ a & \dots & a & a & a & \dots & a & a & a \end{array} \right\}, \\ \\ \left\{ \left\{ a \not{c}\, a \,\$_c\, a \not{c}\, a \,; (a) \right\}, \left\{ (a\, a \dots a) \to (a), (a, a)) \right\} \right\} \end{array} \right\},$$

$$R_2 = \left\{ \begin{array}{l} \left\{ \begin{array}{l} \underline{a \dots a} \;\; \underline{a \dots a} \\ a \dots a \;\; a \dots a \\ \vdots\;\vdots \;\; \vdots, \vdots \;\; \vdots \;\; \vdots \;; a \dots a \\ a \dots a \;\; a \dots a \end{array} \right\}, \\ \\ \left\{ \left\{ a \not{c}\, a \,\$_r\, a \not{c}\, a \,; (a) \right\}, \left\{ \begin{pmatrix} a \\ \vdots \\ a \\ a \end{pmatrix} \to (a), (a, a) \right\} \right\} \end{array} \right\}$$

The computation starts with the initial array in region 1. The permitting mate rule R_1 is applied with the array to itself in region 1. The application of the rules throughout the computation is guided by the permitting array. Due to the type t_{in} of the system, the generated array moves to region 2. In region 2, the permitting mate rule R_2 is applied. As a result, the array of solid square shape is obtained.

The picture language generated by Π_3 consists of solid squares of a's as in Figure 3.

Figure 3 Solid square of $a's$

$$\begin{array}{ccccc} \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} \\ \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} \\ \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} \\ \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} \\ \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} \end{array}$$

Remark: We note that the t-communicating array P system with context-free array rewriting rules generating the set of solid squares of a's given by Subramanian et al. (2009) involves four membranes whereas two membranes are enough when the system is endowed with additional permitting mate operation.

4 Generative power

We determine the generative power of the Permitting mate array P system by comparing with other models.

Figure 4 Array describing pattern T

$$\begin{array}{ccccc} \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} & \boldsymbol{a} \\ x & x & \boldsymbol{a} & x & x \\ x & x & \boldsymbol{a} & x & x \\ x & x & \boldsymbol{a} & x & x \\ x & x & \boldsymbol{a} & x & x \end{array}$$

4.1 Theorem

1 $pAP_2(mate) - EAP_2(REG) \neq \phi$

2 $pAP_2(mate) - AREG \neq \phi$

Proof: The statement 1 can be seen as follows: The picture language generated by Π_4 consisting of T shaped arrays with equal arms is in the family $pAP_2(mate)$ as given below. This picture language Π_4 cannot be generated by a regular array grammar with two membranes (Ceterchi et al., 2003). But the following $pAP_2(mate)$ generates the language Π_4 with exactly two membranes. Consider the permitting mate array P system.

$$\Pi_4 = \left(\{a, x\}, \{a, x\}, \#, [_1[_2]_2]_1], \left\{ \begin{array}{c} a\,a\,a \\ x\,a\,x \\ x\,a\,x \end{array} \right\}, \phi, (R_1, R_2), 2 \right)$$

where

$$R_1 = \left\{ \begin{array}{l} \left\{ \begin{array}{ccc|cccc} a & \ldots & a & a\,a & \ldots & a & a \ldots a \\ x & \ldots & x & a\,x & \ldots & x & x \ldots a \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \quad \vdots \quad \vdots \\ x & \ldots & x & a\,x & \ldots & x & x \ldots a \end{array} , \begin{array}{|ccc} a \ldots a & & a\,a\,a \\ x \ldots x & & x\,a\,x \\ \vdots \quad \vdots \quad \vdots & & \vdots\,\vdots\,\vdots \\ x \ldots x & & x\,a\,x \end{array} ; \ldots, (in) \right\}, \\ \\ \left\{ \begin{array}{l} \left\{ \left\{ a \not{c}\, a \,\$_c\, a \not{c}\, a ; (a\,a\,a) \right\}, \left\{ (a\,a \ldots a\,a \ldots a\,a) \rightarrow (a\,a\,a), (a, a)) \right\} \right\}, \\ \\ \left\{ \left\{ x \not{c}\, a \,\$_c\, a \not{c}\, x ; (x\,a\,x) \right\}, \left\{ (a\,x \ldots x\,x \ldots x\,a) \rightarrow (x\,a\,x), (x, x) \right\} \right\} \end{array} \right\} \end{array} \right\},$$

$$R_2 = \left\{ \begin{array}{l} \left\{ \begin{array}{c} a \ldots a\, a \ldots a \\ \hline x \ldots a\, x \ldots x \\ \vdots\ \vdots \quad \vdots\ \vdots\ \vdots \quad \vdots \\ x \ldots a\, x \ldots x \end{array}, \begin{array}{c} a \ldots a\, a \ldots a \\ \hline x \ldots a\, x \ldots x \\ \vdots \quad \vdots \quad \vdots\ \vdots \quad \vdots \quad \vdots \\ x \ldots a\, x \ldots x \end{array}; \begin{array}{c} x \ldots a\, x \ldots x \\ x \ldots a\, x \ldots x \end{array} \right\}, \\ \left\{ \begin{array}{l} \left\{ \left\{ a \not{c}\, x\, \$_r\, a \not{c}\, x; \begin{pmatrix} x \\ x \end{pmatrix} \right\}, \left\{ \begin{pmatrix} x \\ \vdots \\ x \\ a \end{pmatrix} \to \begin{pmatrix} x \\ x \end{pmatrix}, \left(a, \begin{pmatrix} x \\ x \end{pmatrix} \right) \right\} \right\}, \\ \left\{ \left\{ a \not{c}\, a\, \$_r\, a \not{c}\, a; \begin{pmatrix} a \\ a \end{pmatrix} \right\}, \left\{ \begin{pmatrix} a \\ \vdots \\ a \\ a \end{pmatrix} \to \begin{pmatrix} a \\ a \end{pmatrix}, \left(a, \begin{pmatrix} a \\ a \end{pmatrix} \right) \right\} \right\} \end{array} \right\} \end{array} \right\}$$

The picture language generated by Π_4 consists of T shapes as in Figure 4 where 'x' stands for $empty$.

The statement 2 is due to the fact that no regular array grammar by the nature of its rules that can ensure the arms of equal length. In fact the regular array grammar rules cannot generate two arms together.

4.2 Theorem

$$pAP_2(mate) \cap PCDCFIAGS \neq \phi.$$

Proof: Consider the permitting mate array P system.

$$\Pi_5 = \left(\{0,1\}, \{0,1\}, \#, [_1[_2]_2]_1, \left\{ \begin{matrix} 1\,0 \\ 1\,1 \end{matrix} \right\}, \phi, (R_1, R_2), 2 \right)$$

where

$$R_1 = \left\{ \begin{array}{l} \left\{ \begin{array}{c|ccc|ccc} 1 & 0 \ldots 0 & 1 & & 0 \ldots 0 & 0 & \\ \vdots & \vdots \ldots \vdots & \vdots & & \vdots \ldots \vdots & \vdots & \\ 1 & 0 \ldots 0, & 1 & & 0 \ldots 0; & 0, & (in) \\ \vdots & \vdots \ldots \vdots & \vdots & & \vdots \ldots \vdots & \vdots & \\ 1 & 1 \ldots 1 & 1 & & 1 \ldots 1 & 1 & \end{array} \right\}, \\ \left\{ \begin{array}{l} \{ \{ 1 \not{c}\, 0\, \$_c\, 1 \not{c}\, 0; (0) \}, \{ (0 \ldots 0\, 1) \to (0), (1, 0)) \} \}, \\ \{ \{ 1 \not{c}\, 1\, \$_c\, 1 \not{c}\, 1; (1) \}, \{ (1 \ldots 1\, 1) \to (1), (1, 1) \} \} \end{array} \right\} \end{array} \right\},$$

$$R_2 = \left\{ \begin{array}{l} \left\{ \begin{array}{cccc} 1\,0 \ldots 0 \\ \vdots\;\vdots\;\;\vdots\;\;\vdots \\ 1\,0 \ldots 0 \\ \hline 1\,1 \ldots 1 \end{array} , \begin{array}{cccc} 1\,0 \ldots 0 \\ \vdots\;\vdots\;\;\vdots\;\;\vdots \\ 1\,0 \ldots 0 \\ \hline 1\,1 \ldots 1 \end{array} ; 1\,0 \ldots 0 \right\}, \\ \left\{ \begin{array}{l} \left\{ \{1 \not{c}\, 1 \,\$_r\, 1 \not{c}\, 1; (1)\}, \left\{ \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \rightarrow (1), (1,1) \right\} \right\}, \\ \left\{ \{0 \not{c}\, 1 \,\$_r\, 0 \not{c}\, 1; (0)\}, \left\{ \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \rightarrow (0), (0,0) \right\} \right\} \end{array} \right\} \end{array} \right\}$$

The picture language generated by Π_5 consists of L shapes as in Figure 5 where '0' stands for *empty*. This language also can be generated by $PCDCFIAGS$ (Sheena Christy and Thomas, 2015).

Figure 5 Array describing pattern L

$$\begin{array}{ccccc} \boldsymbol{1} & 0 & 0 & 0 & 0 \\ \boldsymbol{1} & 0 & 0 & 0 & 0 \\ \boldsymbol{1} & 0 & 0 & 0 & 0 \\ \boldsymbol{1} & 0 & 0 & 0 & 0 \\ \boldsymbol{1} & \boldsymbol{1} & \boldsymbol{1} & \boldsymbol{1} & \boldsymbol{1} \end{array}$$

4.3 Theorem

1 $tpAP_1(t_{in}, mate) \subset tpAP_2(t_{in}, mate)$.

2 $CD_2(CFIA, t) \subset tpAP_2(t_{in}, mate)$.

Proof: The proper inclusion in 1 can be seen as follows: the inclusion is proper, since the picture of shape I with equal arms can be generated by $tpAP_2(t_{in}, mate)$, as in Figure 2. But it cannot be generated by t-communicating permitting mate rules in just a single membrane, as the rules cannot maintain equal growth.

The proper inclusion in two can also be proved similarly. The iso-picture of shape L with equal arms can be generated by $CD_3(CFIA, t)$ (Csuhaj-Varju et al., 1994). But we can generate pictures of shape L with equal arms by using $tpAP_2(t_{in}, mate)$, as in Figure 5.

5 Conclusions

We have considered the features of permitting mate operation in the rules and t-communication mode in the regions of the permitting mate array P system. Specific patterns like characters are generated by the array P system with permitting mate rules. It is remarkable to note that the number of membranes are reduced which increases the generative power of $pAP_m(mate)$. The generative power is determined with other models of picture description. It is worth examining the boundedness of the system and the set properties.

References

Busi, N. (2005) 'On the computational power of the mate/bud/drip brane calculus', in *Pre-Proc. Sixth Workshop on Mebrane Computing*, pp.235–252.

Cardelli, L. and Paun, G.H. (2006) 'An universality result for a membrane calculus based on mate/drip operations', *Intern. Journal of Foundations of Computer Science* , Vol. 17, No. 1, pp.49–68.

Cardelli, L.(2005) 'Brane calculi, interactions of biological membranes', in *Proc. Computational Methods in System Biology*, pp.257–280.

Ceterchi, R., Mutyam, M., Paun, G.H. and Subramanian, K.G. (2003) 'Array rewriting P systems', *Natural Computing*, Vol. 2, No. 3, pp.229–249.

Chandra, P.H., Kalavathy, S.M.S.T. and Kalyani, M.N. (2016) 'The computational power of array P system with mate operation', *Bio-inspired Computing Theories and Applications*, Vol. 16, pp.200–214.

Csuhaj-Varju, E., Dassow, J., Kelemen, J. and Paun, G.H. (1994) 'A grammar systems: a grammatical approach to distribution and cooperation', *Topics in Computer Mathematics*, pp.37–49.

Dassow, J. and Paun, G.H. (1989) *Regulated Rewriting in Formal Language Theory*, pp.134–148, Springer-Verlag, Berlin.

Denninghoff, K.L. and Pradalier, S. (1989) 'On the undecidability of splicing systems', *International J. Computer Math*, Vol. 27, Nos. 3–4, pp.133–145.

Ewert, S. and van der Walt, A. (1999) 'Generating pictures using random permitting context', *Int. J. Pattern Recogn. Artificial Intell.*, Vol. 13, No. 3, pp.339–355.

Head, T. (1987) 'Formal language theory and DNA, an analysis of the generative capacity of specific recombinant behaviours', *Bull Math. Biology*, Vol. 49, No. 5, pp.737–759.

Rudolf, F. and Marion, O. (2007) 'Tissue P systems and (mem)brane systems with mate and drip operations working on strings', *Electronic Notes in Theoretical Computer Science*, Vol. 171, No. 2, pp.105–115.

Sheena Christy, D.K. and Thomas, D.G. (2015) 'On the power of permitting features in co-operating iso-array grammar systems', *Int. J. Pure and Applied Mathematics*, Vol. 101, No. 5, pp.663–671.

Subramanian, K.G., Ali, R.M., Nagar, A.K. and Margenstern, M. (2009) 'Array P systems and t-communication', *Fundam. Inform.*, Vol. 91, No. 1, pp.145–159.

Subramanian, K.G., Venkat, I., Pan, L. and Nagar, K.A. (2013) 'Permitting features in P systems and generating picture array', *Advances in Intelligent Systems and Computing*, Vol. 201, pp.27–38.