Available online at www.sciencedirect.com
ScienceDirect
Progress in Natural Science 18 (2008) 611–616
Progress in Natural Science
www.elsevier.com/locate/pnsc

Special report

Trajectory *P* system

Subbaiah Annadurai [a,*], Thiyagarajan Kalyani [a], Vincent Rajkumar Dare [b], Durairaj Gnanaraj Thomas [b]

[a] *Department of Mathematics, St. Joseph's College of Engineering, Chennai 600119, India*
[b] *Department of Mathematics, Madras Christian College, Tambaram, Chennai 600059, India*

Received 21 December 2007; received in revised form 8 January 2008; accepted 10 January 2008

Abstract

Membrane computing is a branch of natural computing aiming to abstract computing ideas for the structure and the functioning of living cells as well as from the way the cells are organized in tissues or higher-order structures. Trajectories are used as a tool for modeling language operations and other related objects. A trajectory *P* system consists of a membrane structure in which the object in each membrane is a collection of words and the evolutionary rules are given in terms of trajectories. In this paper, we present some properties of trajectory *P* systems.

Keywords: Membrane computing; Shuffle trajectories; Parallel computation

1. Introduction

Membrane computing deals with distributed computing models inspired from the structure and the functioning of the living cell [1,2]. Very briefly, compartments (regions) defined by a hierarchical arrangement of membranes have multisets of objects together with evolutionary rules associated with the membranes. Parallel composition of words and languages appears as a fundamental operation in parallel computation and in the theory of concurrency. Usually, this operation is modeled by the shuffle operation or restrictions of this operation, such as literal shuffle and insertion. Roughly speaking, a trajectory is a segment of a line in plane, starting in the origin of axes and continuing parallel with the axis Ox and Oy. The line can change its direction only in points of nonnegative integer coordinates. A trajectory defines how to skip from a word to another word during the shuffle operation.

Shuffle on trajectories [3,4] provides a method of great flexibility to handle the operation of parallel composition of processes from the catenation to the usual shuffle of processes. Also, a membrane serves as a communication channel between a cell and its "neighbors".

This paper brings together two areas of theoretical computer science, namely membrane computing and trajectories, where trajectories are used as evolutionary rules in membrane computing.

In this paper, in Section 3 the algebraic properties of trajectories are studied and in Section 4 we introduce the notion of trajectory *P* system and its properties are discussed.

2. Preliminaries

In this section we deal with the basic concept of *P* system [1,2] and trajectories [3,4].

P system [1] is a new compatibility model of a distributed parallel type based on the notion of a membrane structure. Such a structure consists of computing cells

* Corresponding author. Tel.: +91 44 24501060. fax: +91 44 24500861.
E-mail address: annadurai_70@yahoo.co.in (S. Annadurai).

1002-0071/$ - see front matter © 2008 National Natural Science Foundation of China and Chinese Academy of Sciences. Published by Elsevier Limited and Science in China Press. All rights reserved.
doi:10.1016/j.pnsc.2008.01.003

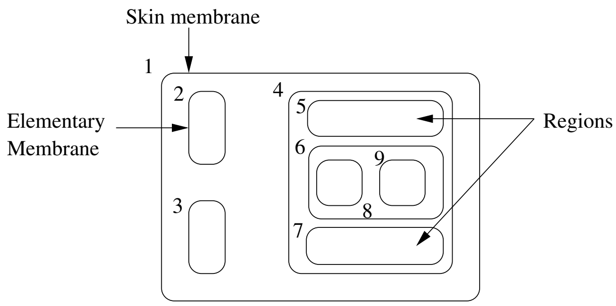

Fig. 1. A membrane structure.

which are organized hierarchically by the inclusion relation. Each cell is enclosed by its membrane. Each cell is an independent computing agent with its own computing program, which produces objects. The interaction between cells consists of the exchange of objects through membranes.

A membrane structure can be represented in a natural way as a Venn diagram (Fig. 1).

The membranes are labeled in a one-to-one manner. Each membrane identifies a region delimited by it and the membranes placed directly inside it (if any). A membrane without any other membrane inside is said to be elementary.

The membrane surrounding the cell which is the highest in the hierarchy is called the skin membrane.

In the regions delimited by the membranes we place multisets of objects from a specified finite set V together with evolutionary rules for these objects.

A P system of degree m, $m \geqslant 1$ is a construct $\pi = (V, T, C, \mu, \mu_1, \mu_2, \dots, \mu_m, (R_1, \rho_1), \dots, (R_m, \rho_m))$where

(i) V is an alphabet; its elements are called objects;
(ii) $T \subseteq V$ (the output alphabet);
(iii) $C \subseteq V$, $C \cap T = \phi$;
(iv) μ is a membrane structure consisting of m membrane;
(v) μ_i, $1 \leqslant i \leqslant m$, are multisets over V associated with the regions $1, 2, \dots, m$ of μ;
(vi) R_i, $1 \leqslant i \leqslant m$, are finite sets of evolutionary rules over V associated with regions $1, 2, \dots, m$ of μ; ρ_i is a partial order relation over R_i, $1 \leqslant i \leqslant m$, specifying a priority relation among rules of R_i. An evolutionary rule is a pair (u, v), which we will usually write in the form $u \to v$, where u is a string over V and $v = v'$ or $v = v'\delta$, where v' is a string over

$$(V \times \{\text{here}, \text{out}\}) \cup (V \times \{\text{in}_j / 1 \leqslant j \leqslant m\}),$$

and δ is a special symbol not in V.

We illustrate the computation of the P system by the following example.

Example 1. Consider the system

$$\pi = (\Sigma, T, C, \mu, w_1, w_2, (R_1, \rho_1), (R_2, \rho_2))$$
$$\Sigma = \{a, b, c, d\},$$

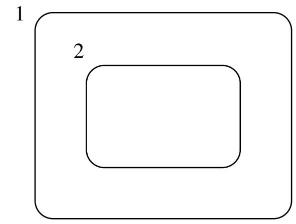

Fig. 2. Example of a P system.

$$T = \{a, c, d\},$$
$$C = \phi,$$
$$\mu = [_1[_2]_2]_1,$$
$$w_1 = a,$$
$$w_2 = \lambda,$$
$$R_1 = \{a \to a(a, \text{out})(b, \text{in}_2),$$
$$a \to b(b, \text{out})(a, \text{in}_2), c \to (c, \text{out}),$$
$$d \to (d, \text{out})\},$$
$$\rho_1 = \phi,$$
$$R_2 = \{b \to bc, a \to d\delta\},$$
$$\rho_2 = \phi.$$

We start working in the skin membrane (see Fig. 2), where a copy of object a is available. By using the rule $a \to a(a, \text{out})(b, \text{in}_2)$, we reproduce the object a in a membrane 1, we send out a copy of the same object, and we introduce a copy of b in membrane 2. From now on, we have applicable rules in both the inner and the outer membranes. At each step in membrane 1. we repeat the previous operations, while in the inner membrane at each step we produce in parallel a copy of c from each available copy of b. For instance, after five steps, we have five copies of a outside, one in membrane 1, five copies of b in membrane 2, and $4 + 3 + 2 + 1 = 10$ copies of c in membrane 2: the first copy of b has produced one c in each of the subsequent step, the next one has evolved only three times, and so on. At any moment, the rule $a \to b(b, \text{out})(a, \text{in}_2)$ can be used in membrane 1. One copy of b is kept in membrane 1, one copy is sent outside (hence the string collected becomes $a^n b$, for some $n \geqslant 0$), and one copy of a is sent to membrane 2. At the same time with the use of the rule $b \to bc$ for all n copies of b present here, we have to use the rule $a \to d\delta$. Membrane 2 is dissolved, its contents are left free in membrane 1, where the rules $c \to (c, \text{out})$ and $d \to (d, \text{out})$ are now applicable. The rules from membrane 2 are no longer available; thus, the $n + 1$ copies of b placed in membrane 1 cannot evolve further. Because the rules $c \to (c, \text{out})$ and $d \to (d, \text{out})$ are used at the same time, in parallel, we get outside the system $n(n + 1)/2$ copies of c and one copy of d. Consequently, any string of the form $a^n bc^i dc^j$, for $n \geqslant 0$ and $i + j = n(n + 1)/2$ belongs to the output of this system. Hence, the language obtained in this way is

$$L(\pi) = \left\{ a^n b c^i d c^j \mid n \geqslant 0, \text{ and } i + j = \frac{n(n+1)}{2}, \quad i, j \geqslant 0 \right\}$$

Definition 1. A trajectory is an element $t \in V^*$, where $V = \{r, u\}$, r and u are versors in the plane; r stands for the right direction, whereas u stands for up direction.

Definition 2. Let Σ be an alphabet and let t be a trajectory, $t = t_1 t_2 \ldots t_n$, where $t_i \in V, 1 \leqslant i \leqslant n$. Let α, β be two words over Σ, $\alpha = a_1 a_2 \ldots a_p, \beta = b_1 b_2 \ldots b_q$, where $a_i, b_j \in \Sigma, 1 \leqslant i \leqslant p$ and $1 \leqslant j \leqslant q$. The shuffle of α with β on the trajectory t, denoted by $\alpha \sqcup\!\sqcup_t \beta$ is defined as follows:

if $|\alpha| \neq |t|_r$ or $|\beta| \neq |t|_u$, then $\alpha \sqcup\!\sqcup_t \beta = \phi$, else $\alpha \sqcup\!\sqcup_t \beta = c_1 c_2 \ldots c_{p+q}$, where, if $|t_1 t_2 \ldots t_{i-1}|_r = k_1$ and $|t_1 t_2 \ldots t_{i-1}|_u = k_2$, then

$$c_i = \begin{cases} a_{k_1+1} & \text{if } t_i = r \\ b_{k_2+1} & \text{if } t_i = u \end{cases}$$

Definition 3. If T is a set of trajectories, where $T \subseteq V^*$ the shuffle of α with β on the set T of trajectories, denoted $\alpha \sqcup\!\sqcup_T \beta$, is

$$\alpha \sqcup\!\sqcup_T \beta = \bigcup_{t \in T} \alpha \sqcup\!\sqcup_t \beta$$

The above operation is extended to languages over Σ, if $L_1, L_2 \subseteq \Sigma^*$, then

$$L_1 \sqcup\!\sqcup_t L_2 = \bigcup_{\alpha \in L_1, \beta \in L_2} \alpha \sqcup\!\sqcup_T \beta$$

Definition 4. A set T of trajectories is commutative if the operation $\sqcup\!\sqcup_T$ is a commutative operation, i.e., $\alpha \sqcup\!\sqcup_T \beta = \beta \sqcup\!\sqcup_T \alpha$, for all $\alpha, \beta \in \Sigma^*$.

Example 2. Let α and β be the two words $\alpha = a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9$, $\beta = b_1 b_2 b_3 b_4 b_5$ and assume that $t = r^3 u^2 r^3 u r u r^2 u$. The shuffle of α with β on the trajectory t is

$$\alpha \sqcup\!\sqcup_t \beta = a_1 a_2 a_3 b_1 b_2 a_4 a_5 a_6 b_3 a_7 b_4 a_8 a_9 b_5$$

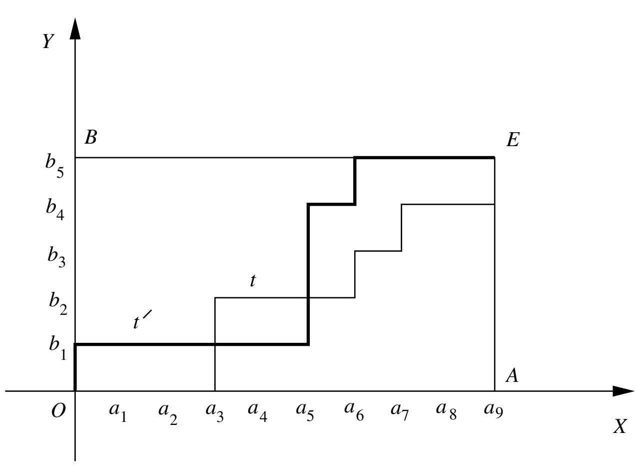

Fig. 3. Geometrical interpretation of trajectories.

The result has the following geometrical interpretation (see Fig. 3): the trajectory t defines a line starting in the origin and continuing one unit to the right or up, depending on the definition of t. Note that the trajectory ends in the point with coordinates $(9, 5)$ that is exactly the upper right corner of the rectangle defined by α and β. Hence, the result of the shuffle of α with β on the trajectory t is nonempty.

Hence, trajectory t defines a line in the rectangle $OAEB$, on which one has "to walk" starting from the corner O, the origin, and ending in the corner E, the exit point. In each lattice point one has to follow one of the versors r or u according to the definition of t.

Now, consider another trajectory t', say $t' = ur^5u^3rur^3$. In Fig. 3, the trajectory t' is depicted by a much thicker line than the trajectory t.

$$\alpha \sqcup\!\sqcup_{t'} \beta = b_1 a_1 a_2 a_3 a_4 a_5 b_2 b_3 b_4 a_6 b_5 a_7 a_8 a_9$$

Consider the set of trajectories $T = \{t, t'\}$. The shuffle of α with β on the set T of trajectories is

$$\alpha \sqcup\!\sqcup_T \beta = \{a_1 a_2 a_3 b_1 b_2 a_4 a_5 a_6 b_3 a_7 b_4 a_8 a_9 b_5, \\ b_1 a_1 a_2 a_3 a_4 a_5 b_2 b_3 b_4 a_6 b_5 a_7 a_8 a_9\}.$$

We have the following theorems appearing in [4].

Theorem 1. *Let T ba a set of trajectories, $T \subseteq \{r, u\}^*$. The following assertions are equivalent*:

(i) For all regular languages $L_1, L_2, L_1 \sqcup\!\sqcup_T L_2$ is a regular language.
(ii) T is a regular language.

Theorem 2. *Let T be a set of trajectories, $T \subseteq \{r, u\}^*$. The following assertions are equivalent*:

(i) For all regular languages $L_1, L_2, L_1 \sqcup\!\sqcup_T L_2$ is a context-free language.
(ii) T is a context-free language.

Theorem 3. *Let T be a set of trajectories, $T \subseteq \{r, u\}^*$ such that T is a regular language.*

(i) If L_1 is a context-free language and if L_2 is a regular language, then $L_1 \sqcup\!\sqcup_T L_2$ is a context-free language.
(ii) If L_1 is a regular language and if L_2 is a context-free language, then $L_1 \sqcup\!\sqcup_T L_2$ is a context-free language.

Theorem 4. *Let L_1, L_2 and T, $T \subseteq \{r, u\}^*$ be three languages.*

(i) If all three languages are regular languages, then $L_1 \sqcup\!\sqcup_T L_2$ is a regular language.

(ii) If two languages are regular languages and the third one is a context-free language, then $L_1 \sqcup\!\sqcup_T L_2$ is a context-free language.

3. Algebraic properties of trajectory

We now provide some algebraic properties of trajectories.

Definition 5. Let L be a language over V and let T be a set of trajectories, then L^T is defined as

$$L^T = \bigcup_{\alpha,\beta \in L} \alpha \sqcup\!\sqcup_T \beta$$

Proposition 1. *Let L, L_1, L_2 be three languages over Σ and $T \subseteq V^*$ be a set of trajectories. Then, we have*

(i) If $L_1 \subset L_2$ then $L_1^T \subseteq L_2^T$.
(ii) $L^{T_1 \cup T_2} = L^{T_1} \cup L^{T_2}$.
(iii) $L^{T_1 \cap T_2} \subseteq L^{T_1} \cap L^{T_2}$.
(iv) $(L_1 \cup L_2)^T = L_1^T \cup L_2^T \cup L_1 \sqcup\!\sqcup_T L_2 \cup L_2 \sqcup\!\sqcup_T L_1$.
(v) $(L_1 \cap L_2)^T \subseteq L_1^T \cap L_2^T$.

Proof

(i) We prove that $L_1 \subseteq L_2 \Rightarrow L_1^T \subseteq L_2^T$. For every $x \in L_1^T$ there exist $y, z \in L_1, t \in T$ such that $x = y \sqcup\!\sqcup_t z$. But we have $y, z \in L_2$ (since $L_1 \subseteq L_2$) with $x = y \sqcup\!\sqcup_T z$, so $x \in L_2^T$ and $L_1^T \subseteq L_2^T$.

(ii) First we prove that, $L^{T_1 \cup T_2} \subseteq L^{T_1} \cup L^{T_2}$.
For every $x \in L^{T_1 \cup T_2}$, there exist $y, z \in L$ and $t \in T_1 \cup T_2$ such that $x = y \sqcup\!\sqcup_t z$.
Since $t \in T_1 \cup T_2$, $t \in T_1$ or $t \in T_2$.
Hence $x \in L^{T_1}$ or $x \in L^{T_2}$ (i.e.,) $x \in L^{T_1} \cup L^{T_2} \Rightarrow L^{T_1 \cup T_2} \subseteq L^{T_1} \cup L^{T_2}$. Now, let $x \in L^{T_1} \cup L^{T_2} \Rightarrow x \in L^{T_1}$ or $x \in L^{T_2}$
$\Rightarrow$ there exist $y, z \in L$ such that $x = y \sqcup\!\sqcup_t z$, $t \in T_1$ or $x = y \sqcup\!\sqcup_t z, t \in T_2$.
$\Rightarrow x = y \sqcup\!\sqcup_t z, t \in T_1 \cup T_2$
$\Rightarrow x \in L^{T_1 \cup T_2}$
(i.e.,) $L^{T_1} \cup L^{T_2} \subseteq L^{T_1 \cup T_2}$
Hence $L^{T_1 \cup T_2} = L^{T_1} \cup L^{T_2}$.

(iii) For every $x \in L^{T_1 \cap T_2}$ there exist $y, z \in L$ and $t \in T_1 \cap T_2$ such that

$$x = y \sqcup\!\sqcup_t z$$
$$\Rightarrow x \in L^{T_1} \text{ and } x \in L^{T_2} \text{ since } t \in T_1 \text{ and } T_2$$
$$\Rightarrow x \in L^{T_1} \cap L^{T_2}$$
$$\Rightarrow L^{T_1 \cap T_2} \subseteq L^{T_1} \cap L^{T_2}$$

(iv) $(L_1 \cup L_2)^T = L_1^T \cup L_2^T \cup L_1 \sqcup\!\sqcup_T L_2 \cup L_2 \sqcup\!\sqcup_T L_1$.
Let $z \in (L_1 \cup L_2)^T$
$(L_1 \cup L_2)^T = \{x \sqcup\!\sqcup_t y / x, y \in L_1 \cup L_2, t \in T\}$
Since $x, y \in L_1 \cup L_2$, $x, y \in L_1$ or $x, y \in L_2$ or $x \in L_1$, $y \in L_2$ or $x \in L_2$, $y \in L_1 \Rightarrow z \in L_1^T$ or $z \in L_2^T$ or $z \in L_1 \sqcup\!\sqcup_T L_2$ or $z \in L_2 \sqcup\!\sqcup_T L_1$
(i.e.,)
$z \in L_1^T \cup L_2^T \cup L_1 \sqcup\!\sqcup_T L_2 \cup L_2 \sqcup\!\sqcup_T L_1 . (L_1 \cup L_2)^T \subseteq L_1^T \cup L_2^T \cup L_1 \sqcup\!\sqcup_T L_2 \cup L_2 \sqcup\!\sqcup_T L_1$.
Let $z \in L_1^T \cup L_2^T \cup L_1 \sqcup\!\sqcup_T L_2 \cup L_2 \sqcup\!\sqcup_T L_1$.
$\Rightarrow z \in L_1^T$ or $z \in L_2^T$ or $z \in L_1 \sqcup\!\sqcup_T L_2$ or $z \in L_2 \sqcup\!\sqcup_T L_1$

$$\Rightarrow z = x \sqcup\!\sqcup_t y, \text{ where } x, y \in L_1 \text{ and } t \in T$$
$$= x \sqcup\!\sqcup_t y, \text{ where } x, y \in L_2 \text{ and } t \in T$$
$$= x \sqcup\!\sqcup_t y, \text{ where } x \in L_1, y \in L_2 \text{ and } t \in T$$
$$= x \sqcup\!\sqcup_t y, \text{ where } x \in L_2, y \in L_1 \text{ and } t \in T$$

$\Rightarrow x, y \in L_1 \cup L_2$
$\Rightarrow z \in (L_1 \cup L_2)^T$ (i.e.,)
$(L_1 \cup L_2)^T = L_1^T \cup L_2^T \cup L_1 \sqcup\!\sqcup_T L_2 \cup L_2 \sqcup\!\sqcup_T L_1$.

(v) $(L_1 \cap L_2)^T = L_1^T \cap L_2^T$
For every $z \in (L_1 \cap L_2)^T$, there exist $x, y \in L_1 \cap L_2$ and $t \in T$ such that

$$z = x \sqcup\!\sqcup_t y$$
$$\Rightarrow x, y \in L_1 \text{ and } x, y \in L_2, t \in T$$
$$\Rightarrow z = x \sqcup\!\sqcup_t y \in L_1^T \text{ and } z = x \sqcup\!\sqcup_t y \in L_2^T$$
$$\Rightarrow z \in L_1^T \text{ and } L_2^T$$
$$\Rightarrow z \in L_1^T \cap L_2^T$$

$(L_1 \cap L_2)^T \subseteq L_1^T \cap L_2^T$. □

Remark 1. Let L be a language over Σ and T_1 and T_2 be trajectories over V, then $L^{T_1 T_2} \neq (L^{T_1})^{T_2}$. For example, consider $L = \{a^n b / n \geqslant 1\}$

$$T_1 = \{r^n u^m / n, m \geqslant 1\}$$
$$T_2 = \{u^n r^m / n, m \geqslant 1|\}$$

Now

$$L^{T_1} = \bigcup_{t \in T_1} x \sqcup\!\sqcup_t y$$
$$= \{a^m b a^n b / m, n \geqslant 1\}$$
$$(L^{T_1})^{T_2} = \{a^m b a^n b a^{m_1} b a^{n_1} b / m, n, m_1, n_1 \geqslant 1\}$$

Similarly $L^{T_1 T_2} = \{a^m a^n bab / m, n \geqslant 1\}$ where $T_1 T_2 = r^n u^m u^n r^m$.
so that $L^{T_1 T_2} \neq (L^{T_1})^{T_2}$.

Proposition 2. *Let L_1, L_2 be languages over Σ and $T \subseteq V^*$ be a set of commutative trajectories. Then, we have $(L_1 \cup L_2)^T = L_1^T \cup L_2^T \cup L_1 \sqcup\!\sqcup_T L_2$.*

Proof. From (iv) of Proposition 1, we have

$$(L_1 \cup L_2)^T = L_1^T \cup L_2^T \cup L_1 \sqcup\!\sqcup_T L_2 \cup L_2 \sqcup\!\sqcup_T L_1.$$

Since T is commutative, we have

$$L_1 \sqcup\!\sqcup_T L_2 = L_2 \sqcup\!\sqcup_T L_1$$
$$(L_1 \cup L_2)^T = L_1^T \cup L_2^T \cup L_1 \sqcup\!\sqcup_T L_2. \quad \square$$

4. Trajectory *P* system

In this section we introduce trajectory P system. For this we require the following notion.

Definition 6. In a membrane structure, if the ith membrane is inside the jth membrane and there is no other membrane containing i inside j, then j is called the immediate successor of i.

For example, in Fig. 1, membrane 1 is the immediate successor of membrane 4 but it is not an immediate successor of membrane 5.

Definition 7. A trajectory P system is defined as
$\pi = (V, T, \mu, \mu_1, \mu_2 \ldots, \mu_n, (L_1, T_1, \mathrm{tar}), (L_2, T_2, \mathrm{tar}), \ldots, (L_n, T_n, \mathrm{tar}))$

where

- (i) V is an alphabet; its elements are called objects;
- (ii) $T = \{r, u\}$ (the control alphabet);
- (iii) μ is a membrane structure consisting of n membranes $\mu_1, \mu_2, \ldots, \mu_n$;
- (iv) $L_i \subset V^*, T_i \subset T^*$ and $\mathrm{tar} \in \{\mathrm{here}, \mathrm{in}_j, \mathrm{out}\}$, $j = 1, \ldots, n$.

For each i, $1 \leqslant i \leqslant n$, let (L_i, T_i) be the content of ith membrane. If i is the elementary membrane then $L_i^{T_i}$ is the language computed in the ith membrane.

The tar = out is attached with all the elementary membranes. Let $L'_i = L_i^{T_i}$ be sent to the immediate successor.

If j is the immediate successor of the ith membrane and if j does not contain any membrane other than i, then $(L'_i \cup L_j)^{T_j}$, computed in the jth membrane and depending on the target attached, is sent to the inner membrane if $\mathrm{tar} = \mathrm{in}_i$ or sent to the outer membrane if tar = out or stays in the same membrane if tar = here.

If j is the immediate successor of m elementary membranes $i_1, i_2, \ldots, i_m$, then the computation is done in each of the m membranes $i_1, i_2, \ldots, i_m$ and the languages $L_{i_1}^{T_{i_1}}, L_{i_2}^{T_{i_2}}, \ldots, L_{i_m}^{T_{i_m}}$ obtained by computations, are sent to the jth membrane. Let $L'_{i_k} = L_{i_k}^{T_{i_k}}, 1 \leqslant k \leqslant m$. Then the computation is done in the jth membrane and the language obtained is $L'_j = (L'_{i_1} \cup L'_{i_2} \cup \ldots \cup L'_{i_m} \cup L_j)^{T_j}$, depending on the target attached. It is either sent to any one of the m elementary membranes if target is $\mathrm{tar} = \mathrm{in}_{i_k}, 1 \leqslant k \leqslant m$, or sent to the outer membrane if the target tar = out or stays in the membrane if tar = here.

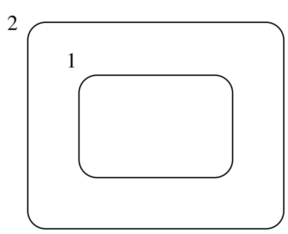

Fig. 4. Membrane structure of a trajectory P system I.

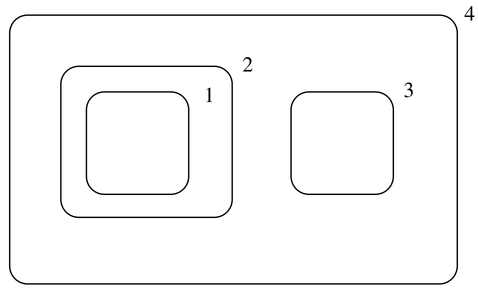

Fig. 5. Membrane structure of trajectory P system II.

This process is repeated till the language is sent to the skin membrane and the language obtained by the computations in the skin membrane is denoted by $L(\pi)$.

Example 3. Consider the system $\pi = (V, T, \mu, \mu_1, \mu_2, (L_1, T_1, \mathrm{out}), (L_2, T_2, \mathrm{in}_1))$ where $V = \{a, b\}, T = \{r, u\}$ and the membrane structure (Fig. 4)

Let

$$L_1 = \{a^2, b^2\};\ T_1 = \{r^n u^n / n \geqslant 1\}$$
$$L_2 = \{b^2\};\ T_2 = \{r^n u^m r^n / m, n \geqslant 1\}$$
$$L'_1 = L_1^{T_1} = \{a^4, b^4, a^2b^2, b^2a^2\}$$
$$L'_2 = (L'_1 \cup L_2)^{T_2} = \{a^2b^4, a^2b^2a^2, b^2b^2a^2, b^6\}$$
$$L''_1 = (L'_2 \cup L_1)^{T_1} = \{a^2b^4a^2b^2a^2, a^2b^4b^2b^2a^2, a^2b^4b^6, a^2b^2a^2b^2b^2a^2, a^2b^2a^2b^6, b^2b^2a^2b^6, a^2b^2\}$$
$$L''_2 = (L_2 \cup L''_1)^{T_2} = \{a^2b^2b^2, a^2b^4a^2b^4b^2b^2a^2a^2b^2a^2, a^2b^4a^2b^4b^6a^2b^2a^2, a^2b^4a^2b^2a^2b^2b^2a^2a^2b^2a^2, \ldots\}$$
$$L(\pi) = (L_1^n \cup L_2)^{T_2} = \left\{ x_1yx_2 \Big/ \begin{array}{l} x = x_1x_2 \\ x, y \in L_1^n \cup L_2 \\ |x_1| = |x_2| \end{array} \right\}$$

$L(\pi)$ is a context-free language.

Example 4. Consider the system $\pi = (V, T, \mu, \mu_1, \mu_2, \ldots, \mu_4, (L_1, T_1, \mathrm{out}), (L_2, T_2, \mathrm{in}_1), (L_3, T_3, \mathrm{out}), (L_4, T_4, \mathrm{in}_i) i = 2, 3)$ where $V = \{a, b\}, T = \{r, u\}$ and membrane structure as given in Fig. 5. Let

$$L_1 = \{a^n / n \geqslant 1\};\ T_1 = \{r^n u^m / m, n \geqslant 1\}$$
$$L_2 = \{b^n / n \geqslant 1\};\ T_2 = \{r^n u / n \geqslant 1\}$$
$$L_3 = \{ab^{n+1} / n \geqslant 1\};\ T_3 = \{r^m u^n / m, n \geqslant 1\}$$
$$L_4 = \{ba^{n+1} / n \geqslant 1\};\ T_4 = \{u^n r^n / n \geqslant 1\}$$
$$L'_1 = L_1^{T_1} = \{a^{n+m} / n, m \geqslant 1\}$$
$$L'_2 = (L'_1 \cup L_2)^{T_2} = \{a^{n+m}b, b^na, a^{m+n+1}, b^{n+1} / n, m \geqslant 1\}$$
$$L''_1 = (L'_2 \cup L_1)^{T_1} = \{a^{n+m}bb^na, a^{n+m}ba^{m+n+1}, a^{n+m}bb^{n+1}, \ldots / n, m \geqslant 1\}$$

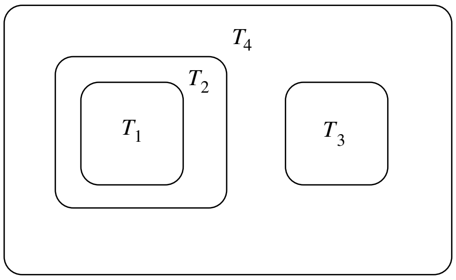

Fig. 6. Membrane structure of trajectory P system III.

$$
\begin{aligned}
L_2'' &= (L_1'' \cup L_2)^{T_2} \\
&= \{a^{n+m}bb^nab, a^{n+m}ba^{m+n+1}b, \ldots / n, m \geqslant 1\} \\
L_3' &= L_3^{T_3} = \{ab^n, ab^m / n, m \geqslant 1\} \\
L_4' &= (L_2'' \cup L_3' \cup L_4)^{T_4} \\
&= \{a^{n+m}bb^nabab^{2n+m+2}, \\
&\quad a^{n+m}ba^{m+n+1}bab^{2n+2m+2}, \ldots / n, m \geqslant 1\}
\end{aligned}
$$

$L(\pi)$ is any one of the languages in L_4' which is a context-free language.

Remark 2. In a trajectory P system $\pi = (V, T, \mu, \mu_1, \mu_2, \ldots, \mu_4, (L_1, T_1, \text{out}), (L_2, T_2, \text{out}), \ldots, (L_4, T_4, \text{out}))$, with membrane structure as given in Fig. 6. If $L_1 = L$, $L_2 = \phi$, $L_3 = L$ and $L_4 = \phi$, then

$$
\begin{aligned}
L(\pi) =& ((L^{T_1})^{T_2} \cup L^{T_3})^{T_4} \\
=& ((L^{T_1})^{T_2})^{T_4} \cup (L^{T_3})^{T_4} \cup \\
& (L^{T_1})^{T_2} \sqcup\!\sqcup_{T_4} L^{T_3} \cup L^{T_3} \sqcup\!\sqcup_{T_4} (L^{T_1})^{T_2}
\end{aligned}
$$

Theorem 5. *In a trajectory P system* $\pi = (V, T, \mu, \mu_1, \mu_2, \ldots, \mu_n, (L_1, T_1, \text{tar}), (L_2, T_2, \text{tar}), \ldots, (L_n, T_n, \text{tar}))$. *Suppose* T_1, $T_2, \ldots, T_n$ are regular languages, then for all regular languages $L_1, L_2, \ldots, L_n$, $L(\pi)$ *is a regular language.*

Proof. From the definition of trajectory P system if tar = out for all membranes, we have

$$L_j' = [((L_k^{T_k} \cup L_{k+1})^{T_{k+1}} \cup L_{k+2})^{T_{k+2}} \cup L_r]^{T_r}$$

It follows from the Theorem 1 that L_j' is regular for all $j = 1, 2, \ldots, m$.

$L_1' \cup L_2' \cup \cdots \cup L_m'$ is regular, since each L_j' is regular. By Theorem 1, it follows that

$$L(\pi) = ((L_1' \cup L_2' \cup \cdots \cup L_m') \cup L_n)^{T_n}$$

is regular.

In general, we obtain that $L(\pi)$ is regular if tar $\in \{\text{here}, \text{in}_j, \text{out}\}$. □

Theorem 6. *In a trajectory P system* $\pi = (V, T, \mu, \mu_1, \mu_2, \ldots, \mu_n, (L_1, T_1, \text{tar}), (L_2, T_2, \text{tar}), \ldots, (L_n, T_n, \text{tar}))$. *Suppose* $T_1, T_2, \ldots, T_n$ *are context-free languages, then for all regular languages* $L_1, L_2, \ldots, L_n$, $L(\pi)$ *is a context-free language.*

Proof. The proof is similar to the proof of Theorem 5 and it follows from Theorem 2. □

5. Conclusion

In this paper, we have introduced trajectory P system and studied its properties.

Acknowledgment

The authors thank Prof. Gh. Păun for his suggestions.

References

[1] Păun Gh. Computing with membranes. J Comput Syst Sci 1998;61(1):108–43.
[2] Păun Gh, Rozenberg G, Salomaa A. Membrane computing with external output. Fundam Inform 2000;34:313–40.
[3] Ahmad Kadrie, Dare VR, Thomas DG, et al. Algebraic properties of the shuffle over ω-trajectories. Inform Process Lett 2001;80:139–44.
[4] Mateescu A, Rozenberg G, Salomaa A. Shuffle on trajectories: syntactic constraints. Theor Comput Sci 1998;197:1–56.