A New Molecular Computing Model, Artificial Cell Systems

Yasuhiro Suzuki and Hiroshi Tanaka
Department of Bio-Informatics
Medical Research Institute,
Tokyo Medical and Dental University
Yushima 1-5-45, Bunkyo, Tokyo 113 JAPAN

Abstract

We develop a new molecular computing model, Artificial Cell System (ACS). ACS consists of a multiset of symbols, a set of rewriting rules (*reaction rules*) and membranes. Throughout a simulation, we find that cells evolve to a structure consisting of several cell-like membranes. We investigate the correlation between the type of reaction rules and the evolution of a cell and find a parameter to describe the correlation.

1 Introduction

Membrane is an important structure for living systems. It distinguishes "self" from its environment and composes hierarchical structures inside the system (like cells, organs and so on). We harness membrane to construct a computing model. This contribution is a preliminary research for it. We will propose two types of systems and a parameter which describe the behavior of the membrane structure.

We have developed an abstract computational model (ARMS), which can deal with systems with many degrees of freedom and confirm that it can simulate the chemical oscillations that are often found in the emergence of life.

Based on this system, we developed an Artificial Cell System (ACS). It consists of a multiset of symbols, a set of rewriting rules (*reaction rules*) and membranes.

2 *ARMS*

We will introduce the multiset rewriting system, "*Abstract Rewriting system on MultiSets*" in this section. Intuitively, $ARMS$ is like a chemical solution in which *molecules* floating on it can interact with each other according to reaction rules. Technically, a chemical solution is a finite multi-set of elements denoted by $A^k = \{a, b, \ldots, \}$; these elements correspond to *molecules*. Reaction rules that act on the molecules are specified in $ARMS$ by rewriting rules. As to the intuitive meaning of $ARMS$, we refer to the study of chemical abstract machines [2]. In fact, this system can be thought of as an underling "*algorithmic chemistry* [1]."

Let A be an *alphabet* (a finite set of abstract symbols). The set of all strings over A is denoted by A^*; the empty string is denoted by λ. (Thus, A^* is the free monoid generated by A under the operation of concatenation and having the identify λ.) The length of a string $w \in A^*$ is denoted by $|w|$.

A *rewriting rule* over A is a pair of strings (u, v), $u, v \in A^*$. We write such a rule in the form $u \to v$. Note that u and v can also be empty. A *rewriting system* is a pair (A, R), where A is an alphabet and R is a finite set of rewriting rules over A.

With respect to a rewriting system $\gamma = (A, R)$ we define over A^* a relation $\Longrightarrow$ as follows: $x \Longrightarrow y$ iff $x = x_1 u x_2$ and $y = x_1 v x_2$, for some $x_1, x_2 \in A^*$ and $u \to v \in R$. The reflexive and transitive closure of this relation is denoted by $\Longrightarrow^*$. A string $x \in A^*$ for which there is no string $y \in A^*$ such that $x \Longrightarrow y$ is said to be an *dead* one (in other words, from a dead string no string can be derived by means of the rewriting rules).

From now on, we work with an alphabet A whose elements are called *objects*; the alphabet itself is called a *set of objects*.

A *multiset* over a set of objects A is a mapping $M : A \longrightarrow \mathbf{N}$, where $\mathbf{N}$ is the set of natural numbers, 0, 1, 2,.... The number $M(a)$, for $a \in A$, is the *multiplicity* of object a in the multiset M. Note that we do not accept here an infinite multiplicity.

We denote by $A^{\#}$ the set of all multisets over A, including the empty multiset, $\emptyset$, defined by $\emptyset(a) = 0$ for all $a \in A$.

A multiset $M : A \longrightarrow \mathbf{N}$, for $A = \{a_1, \ldots, a_n\}$, can be naturally represented by the string $a_1^{M(a_1)} a_2^{M(a_2)} \ldots a_n^{M(a_n)}$ and by any other permutation of this string. Conversely, with any string w over A we can associate a multiset: denote by $|w|_{a_i}$ the number of occurrences of object a_i in w, $1 \leq i \leq n$; then, the multiset associated with w, denoted by M_w, is defined by $M_w(a_i) = |w|_{a_i}, 1 \leq i \leq n$.

The union of two multisets $M_1, M_2 : A \longrightarrow \mathbf{N}$ is the multiset $(M_1 \cup M_2) : A \longrightarrow \mathbf{N}$ defined by $(M_1 \cup M_2)(a) = M_1(a) + M_2(a)$, for all $a \in A$. If $M_1(a) \leq M_2(a)$ for all $a \in A$, then we say that multiset M_1 is included in multiset M_2 and we write $M_1 \subseteq M_2$. In such a case, we define the multiset difference $M_1 - M_2$ by $(M_2 - M_1)(a) = M_2(a) - M_1(a)$, for all $a \in A$. (Note that when M_1 is not included in M_2, the difference is not defined).

A rewriting rule such as

$$a \rightarrow a \ldots b,$$

is called a *heating rule* and denoted as $r_{\Delta > 0}$; it is intended to contribute to the stirring solution. It breaks up a complex *molecule* into smaller ones: *ions*. On the other hand, a rule such as

$$a \ldots c \rightarrow b,$$

is called a *cooling rule* and denoted as $r_{\Delta < 0}$; it rebuilds *molecules* from smaller ones. In this paper, reversible reactions, i.e., $S \rightleftharpoons T$, are not considered. We shall not formally introduce the refinement of *ions* and *molecules* though we use refinement informally to help intuition (on both types of rules we referred to [2]).

A *multiset rewriting rule* (we also use to say, *evolution rule*) over a set A of objects is a pair (M_1, M_2), of elements in $A^{\#}$ (which can be represented as a rewriting rule $w_1 \rightarrow w_2$, for two strings $w_1, w_2 \in A^*$ such that $M_{w_1} = M_1$ and $M_{w_2} = M_2$). We use to represent such a rule in the form $M_1 \rightarrow M_2$.

An *abstract rewriting system on multisets* (in short, an *ARMS*) is a pair

$$\Gamma = (A, (R, \rho))$$

where:

(1) A is a set of objects;

(2) R is a finite set of multiset evolution rules over A;

(3) ρ is a partial order relation over R, specifying a *priority* relation among rules of R.

With respect to an *ARMS* Γ, we can define over $A^{\#}$ a relation: $(\Longrightarrow)$: for $M, M' \in A^{\#}$ we write $M \Longrightarrow M'$ iff

$$M' = (M - (M_1 \cup \ldots \cup M_k)) \cup (M_1' \cup \ldots \cup M_k',)$$

for some $M_i \rightarrow M_i' \in R, 1 \leq i \leq k, k \geq 1$, and there is no rule $M_s \rightarrow M_s' \in R$ such that $M_s \subseteq (M - (M_1 \cup \ldots \cup M_k))$; at most one of the multisets $M_i, 1 \leq i \leq k$, may be empty.

With respect to an *ARMS* $\Gamma = (A, R)$ we can define various types of multisets:

- A multiset $M \in A^{\#}$ is *dead* if there is no $M' \in A^{\#}$ such that $M \Longrightarrow M'$ (this is equivalent to the fact that there is no rule $M_1 \rightarrow M_2 \in R$ such that $M_1 \subseteq M$).
- A multiset $M \in A^{\#}$ is *initial* if there is no $M' \in A^{\#}$ such that $M' \Longrightarrow M$.

2.1 How *ARMS* works

Example In this example, an *ARMS* is defined as follow;

$$\begin{array}{lcl} \Gamma & = & (A, (R, \rho)), \\ A & = & \{a, b, c, d, e, f\}, \\ R & = & \{a, a, a \rightarrow c : r_1, b \rightarrow d : r_2, c \rightarrow e : r_3, \\ & & d \rightarrow f, f : r_4, a \rightarrow a, b, b, a : r_5, f \rightarrow h : r_6, \}. \\ \rho & = & \emptyset \end{array}$$

The set of the rewriting rules, R is $\{r_1, r_2, r_3, r_4, r_5, r_6\}$, we do not assume *priority* among these rules. We assume the maximal multiset size is 4 and the initial state is given by $\{a, a, b, a\}$. In *ARMS*, reaction rules are applied in parallel. When there are more than two applicable-rules then one rule is selected randomly. Figure 1 illustrates an example of rewriting steps of the calculation from the initial state.

At the first step, the left hand side of rule of r_1, r_2 and r_5 are included in the initial state. Since, when it uses r_5 the size of multiset exceeds the maximal size, it uses r_1 and r_2 in parallel. In the next step, r_3 and r_4 are applied in parallel and $\{c, d\}$ is rewritten into

$\{a,b,a,a\}$ $\subseteq a,\ a,\ a,\ a,\ b$ (the left hand side of r_1, r_2, r_5)

$\downarrow$

$\{c,d\}$ $\subseteq c,\ d$ (the left hand side of r_3, r_4)

$\downarrow$

$\{e,f,f\}$ $\subseteq f$ (the left hand side of r_6)

$\downarrow$

$\{e,h,h\}$ There are no rule to apply, it reaches the *death* state

Figure 1: Example of rewriting steps of $ARMS$

$\{e,f,f\}$. In step 3, by using r_6, $\{e,f,f\}$ is transformed into $\{e,h,h\}$. There are no rules that can transform the multiset any further so, the multiset is in a *dead* state.

Multiplication by using $ARMS$

By using $ARMS$, we can do the multiplication of $m \times n$. The $ARMS$ for multiplication is defined as follows;

$$\begin{aligned} \Gamma &= (A,(R,\rho)), \\ A &= \{a,b,c,d,\}, \\ R &= \{a,c \rightarrow a,b : r_1, b^m d \rightarrow b^m c^m : r_2\}, \\ \rho &= r_1 > r_2, \end{aligned}$$

where m in rule r_2 is a parameter denoting the first multiplicant. For example, to calculate 2×3, r_2 is applied as $b,b,d \rightarrow b,b,c,c$.

The initial state of the calculation is defined as $\{a^m, c^m, d^{n-1}\}$; for example, to calculate 2×3, it is $\{a,a,c,c,d,d\}$.

In the initial state, r_1 is applied in parallel and $\{a,a,c,c,d,d\}$ is transformed into $\{a,a,b,b,d,d\}$. Since r_1 can not be applied on this multiset r_2 is applied to it next, resulting in the multiset $\{a,a,b,b,c,c,d\}$. For this multiset r_1 is applied again in parallel and the multiset becomes $\{a,a,b,b,b,b,d\}$. Then r_2 is used again, resulting into $\{a,a,b,b,b,b,c,c\}$. After applying r_1 again, the multiset is transformed into $\{a,a,b,b,b,b,b,b\}$. For this multiset there are no rules that apply, thus it is a *dead* state. The answer of this calculation is obtained as $M(b)$ and in this case it is 6 (figure 2). Addition, subtraction and division can also be implemented on $ARMS$ easily.

$\{a,a,c,c,d,d\}$

$\downarrow$ .... r_1

$\{a,a,b,b,d,d\}$

$\downarrow$ .... r_2

$\{a,a,b,b,c,c,d\}$

$\downarrow$ .... r_1

$\{a,a,b,b,b,b,d\}$

$\downarrow$ .... r_2

$\{a,a,b,b,b,b,c,c\}$

$\downarrow$ .... r_1

$\{a,a,b,b,b,b,b,b\}$ .... *dead* state

Figure 2: Multiplication 2×3 on an $ARMS$

3 Artificial Cell System

In this section, we introduce the basic structural ingredients of ARMS, membrane structures and how ACS works.

3.1 The membrane structure (MS)

To describe the membrane and its structure in $ARMS$ (see [7]), we first define the language MS over the alphabet $\{[,]\}$, whose strings are recurrently defined as follows:

(1) $[,] \in MS$

(2) if $\mu_1, ..., \mu_n \in MS$, n ≥ 1, then $[\mu_1, ..., \mu_n] \in MS$;

(3) there is nothing else in MS.

Consider now the following relation on MS: for $x, y \in MS$ we write $x \sim y$ if and only if we can write the two strings in the form $x = [_1 ... [_2 ...]_2 [_3 ...]_3 ...]_1, y = [_1 ... [_3 ...]_3 [_2 ...]_2 ...]$, i.e, if and only if two pairs of parentheses which are not contained in one another can be interchanged, together with their contents. We also denote by $\sim$ the reflexive and transitive closures of the relation $\sim$. This is clearly an equivalence relation. We denote by $\overline{MS}$ the set of equivalence classes of MS with respect to this relation. The elements of $\overline{MS}$ are called *membrane structures.*

It is easy to see that the parentheses [,] appearing in a membrane structure are matching correctly in the usual sense. Conversely, any string of correctly matching pairs of parentheses [,], with a matching pair at the ends, corresponds to a membrane structure.

Each matching pair of parentheses [,] appearing in a membrane structure is called a *membrane*. The number of membranes in a membrane structure μ is called the *degree* of μ and is denoted by $\deg(\mu)$. The external membrane of a membrane structure μ is called the *skin* membrane of μ. When a membrane which appears in $\mu \in \overline{MS}$ has the form [ ] and no other membrane appear inside the two parentheses then it is called an *elementary* membrane.

3.2 Evolution of ACS

A transition ACS is a construct

$$\Gamma = (A, \mu, M_1, ..., M_n, (R, \rho), MC, \delta, \sigma),$$

where:

(1) A is a set of objects;

(2) μ is a membrane structure (it can be changed throughout a computation);

(3) $M_1, ..., M_n$, are multisets associated with the regions 1,2, ... n of μ;

(4) R is a finite set of multiset evolution rules over A and ρ is a partial order relation over the rule set, it specifies a *priority* relation among the rules.

(5) MC is a set of membrane compounds;

(6) δ is the threshold value of dissolving membrane;

(7) σ is the threshold value of dividing membrane;

μ is a membrane structure of degree n, n $\geq$ 1, with the membranes labeled in a one-to-one manner, for instance, with the numbers from 1 to n. In this way, also the regions of μ are identified by the numbers from 1 to n.

Reaction rules are applied in following manner:

(1) The same rules are applied to every membrane. There are no rules specific to a membrane.

(2) All the rules are applied in parallel. In every step, all the rules are applied to all objects in every membrane that can be applied. If there are more than two rules that can apply to an object then one rule is selected randomly.

(3) If a membrane dissolves, then all the objects in its region are left free in the region immediately above it.

(4) All objects and membranes not specified in a rule and which do not evolve are passed unchanged to the next step.

We have developed two types of ACS as follows;

(1) ACS with Elementary membrane (ACSE); in this system every membrane is elementary membrane.

(2) ACS with Hierarchically structure-able membrane (ACSH); In this system other membranes can appear inside a membrane.

3.3 ACS with Elementary membrane (ACSE)

We will address ACSE first.

3.3.1 Dissolving and dividing a membrane

A membrane is composed of a "membrane compound" which is in fact a symbol. To maintain a membrane, it needs to have a certain minimal volume. A membrane disappear if the volume of membrane compounds decreases below the needed volume to maintain the membrane. Dissolving the membrane is defined as follows:

$$[_i a, b, ...]_i \rightarrow a, b, ...$$

where the ellipsis {...} illustrates chemical compounds inside the membrane. Dissolving takes place when

$$\frac{|w_i|_{MC}}{|M_i|} < \delta$$

where δ is a threshold value when a membrane is dissolved, all chemical compounds in its region are set free and they are merged into the region immediately above it.

On the other hand, when the volume of membrane compounds increases to a certain extend, a membrane is divided. Dividing a membrane is realized by dividing it on a multisets of the random sizes. The frequency at which a membrane is divided is decided in proportion to its size. As the size of a multiset becomes larger, the cell is divided more frequently. Technically, this is defined as follows;

$$[_h a, b, ...]_h \rightarrow [_i a, ...]_i, [_j b, ...]_j$$

Dividing takes place when

$$\frac{|w_h|_{MC}}{|M_h|} > \sigma$$

where σ is a threshold when a membrane is divided, all chemical compounds in its region are set free and they are separated randomly by new membranes.

Setting of a simulation

We assume that there are some cells in the initial state. Some of them are absorbed to a cell and a cell disposes arbitrary chemical compounds to the soup, vice versa. We do not assume that characteristics of a membrane, thus a cell absorbs and disposes arbitrary selected compounds. The number of chemical compounds in the system is kept at the same.

Rule set The length of the left- or right-hand-side of a rule is between one and seven. Both sides of the rules are obtained by sampling with replacement of two symbols a and b. Thus the left/right hand side of a rule is a or $a, a, or ... a^7$ or b or, b, b or $...b^7$ or $a^n, b^m (2 < n + m < 7)$ and the set of reaction rules is constructed as the overall permutation of them. When a rule is applied it is selected randomly from the set.

Cell like Chemoton

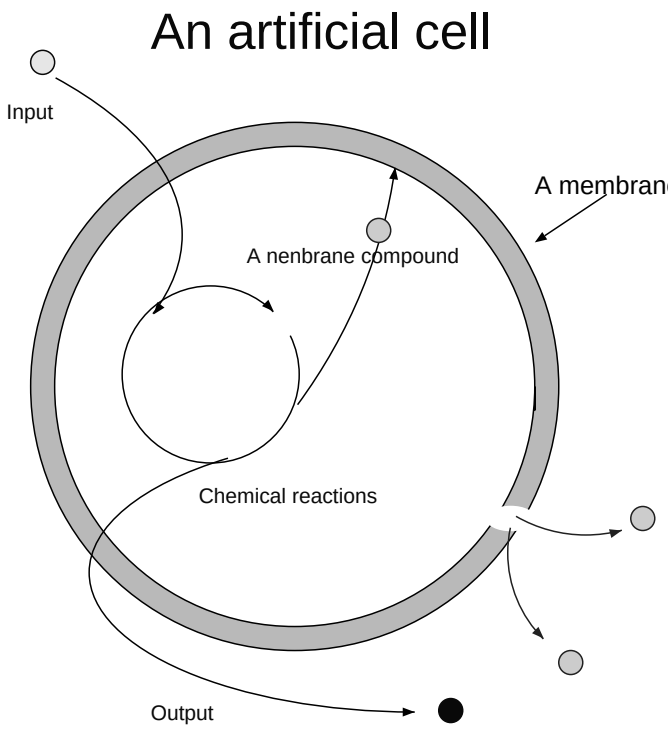

Figure 3: An artificial cell in a *ACSE*

Because of the components of a membrane diminish with the lapse of a certain time, a cell has to generate the components to maintain the membrane through chemical reaction in the cell. Hence, survived cells become cells like *chemoton*[4] (figure 2). We confirmed it through a simulation under the setting.

Cell Cycle

In a simulation, we found that behavior like *cell cycle* emerges. The cycle emerges as follows;

(1) **Dividing period**; Large cells divide into small cells and the number of cells increase.

(2) **Growing period**; Each cell grows larger and the number of cells decreases. Since the resources, the total of compounds in the system are kept at the same, the growing of each cell stops at a certain extent and shifts to dividing period.

3.4 ACS with Hierarchically structure-able membrane (ACSH)

3.4.1 Dissolving and dividing a membrane

A membrane is composed of a "membrane compound" which is in fact a symbol. To maintain a membrane, it needs to have a certain minimal volume. A membrane disappear if the volume of membrane compounds decreases below the needed volume to maintain the membrane. Dissolving the membrane is defined as follows:

$$[_h a, ...[_i b, ...]_i]_h \rightarrow [_h a, b, ...]_h ,$$

where the ellipsis $\{...\}$ illustrates chemical compounds inside the membrane. Dissolving takes place when

$$\frac{|w_i|_{MC}}{|M_i|} < \delta,$$

where δ is a threshold value when a membrane is dissolved, all chemical compounds in its region are set free and they are merged into the region immediately above it.

On the other hand, when the volume of membrane compounds increases to a certain extend, a membrane is divided. Dividing a membrane is realized by dividing it on a multisets of the random sizes. The frequency at which a membrane is divided is decided in proportion to its size. As the size of a multiset becomes larger, the cell is divided more frequently. Technically, this is defined as follows;

$$[_h a, b, ...]_h \rightarrow [_h a, ...[_i b, ...]_i]_h$$

Dividing takes place when

$$\frac{|w_h|_{MC}}{|M_h|} > \sigma$$

where σ is a threshold when a membrane is divided, all chemical compounds in its region are set free and they are separated randomly by new membranes.

4 Behavior of *ACSH*

We implemented the system and did some experiments. We will show some experimental results in this section.

Description of a simulation

The following *ACSH* was simulated;

$$\Gamma = (A = \{a, b, c\}, \mu = \{\emptyset\}, M_0 = \{a^{10}, b^{10}, c^{10}\},$$
$$(R, \rho = \{\emptyset\}), MC = \{b\}, \delta = 0.4, \sigma = 0.2),$$

where:

(1) R is a finite set of multiset evolution rules over A. The length of the left- or right-hand-side of a rule is between one and three. Both sides of the rules are obtained by sampling with replacement of three symbols a, b and c. A set of reaction rules is constructed as the overall permutation of both sides of the rules. We do not assume partial order relations over the rule set in this simulation $\rho = \{\emptyset\}$;

(2) Membrane structures are assumed to have a nonempty ($\mu = \{\emptyset\}$).

As in ACSE, because the components of a membrane diminish with the lapse of a certain time, a cell has to generate the components to maintain the membrane by chemical reactions in the cell. Hence, "surviving" cells become cells like a *chemoton* [4] (figure 3).

In order to investigate the correlation between the reaction rule and the behavior of the model, we will introduce the λ_e parameter [13]. This parameter is introduced as an order parameter of *ARMS*.

Let us define the λ_e parameter as follows:

$$\lambda_e = \frac{\Sigma r_{\Delta S>0}}{1 + (\Sigma r_{\Delta S<0} - 1)} \qquad (1)$$

where $\Sigma r_{\Delta S>0}$ corresponds to the number of *heating rules*, and $\Sigma r_{\Delta S<0}$ to the number of *cooling rules*. This parameter is well-defined when the number of rules is greater than 1.

When the *ARMS* only uses rules of the type $r_{\Delta S<0}$, λ_e is equal to 0.0. On the contrary, if the *ARMS* uses rules of the type $r_{\Delta S>0}$ and $r_{\Delta S<0}$ with the same frequency, λ_e is equal to 1.0. Finally, when the *ARMS* only uses rules of the type $r_{\Delta S>0}$, λ_e is greater than 1.0.

λ_e indicates the degree of reproduction in a cell. When λ_e close to 0.0, the degree is quiet low and as λ_e is getting larger than 1.0, the degree becomes high.

λ_e parameter and system's behavior The behavior of *ACSH* is classified into four classes by this parameter as follows;

- Type I: A cell dose not evolve and disappears;
- Type II: The period of dividing membranes and dissolving membranes appears cyclically, can be seen.
- Type III: A cell evolves to a complex, hierarchically structured cell;
- Type IV: All chemical compounds inside a cell increase rapidly, however, cells hardly divide;

where each λ_e value is not important very much. Although they are changed under the different environments, these four classes are unchanged.

Type I (λ_e closes to 0.0)

When λ_e is close to 0.0 the *cooling rule* is mainly used, and cells will hardly grow up. Thus membranes will hardly to be divide (figure 4).

Type II (λ_e in between 0.5 and $\sim$ 1.0)

When λ_e is in between 0.5 $\sim$ 1.0, membranes are more likely to be divided than when λ_e is close to 0.0. But, since *cooling rules* are likely to be used, the membrane compounds do not increase very much. Thus, when a cell evolves to a certain size the membrane compounds of each cell decrease and they are dissolved (figure 5).

step	state
0.	$[a^{10}, b^{10}, c^{10}]$
1.	$[a^{9}, b^{5}, c^{10}]$
2.	$[a^{10}, b^{2}, c^{7}]$
3.	$[a^{11}, b^{3}, c^{7}]$
4.	$[a^{6}, b^{4}, c^{5}]$
	.....
10.	$[a^{1}, b^{4}, c^{2}]$
	.....
16.	$[a^{1}, b^{4}]$

Figure 4: An example of state transition of *ACSH*: Type I (λ_e closes to 0.0)

Type III (λ_e in between 1.05 $\sim$ 2.33)

When λ_e exceeds 1.0 and the *heating rule* is likely to be used, a cell grows up easier and the frequency of cell division becomes high. Thus a cell evolves into a large complex cell (figure 6).

Type IV (λ_e more than 2.5)

When the λ_e parameter becomes much larger than 1.0, membranes will hardly be divided, because, the number of compounds of all kinds in the cell increase and

step	state
0.	$[a^{10}, b^{10}, c^{10}]$
1.	$[a^{6}, b^{9}, c^{9}]$
	.....
91.	$[a^{4}, b^{4}, c^{1}]$
92.	$[[b^{2}]\ [b^{3}, c^{1}]]$
93.	$[[b^{2}]\ [b^{1}, c^{1}]]$
94.	$[a^{6}, b^{1}, c^{1}]$
	.....
114.	$[a^{2}, b^{5}]$
115.	$[[a^{1}, b^{2}][a^{1}, b^{3}]]$
116.	$[[\ a^{1}, b^{2}][b^{2}][a^{1}, b^{1}]]$
	.....
140.	$[[[\ a^{1}, b^{1}\ ][b^{1}, c^{1}]][[b^{2}\ ][a^{1}, b^{1}]][b^{1}, c^{1}]]$
141.	$[[a^{3}, c^{3}][a^{3}, c^{2}][a^{1}, c^{2}]$
142.	$[\ a^{6}, c^{5}\ ]$

Figure 5: An example of state transition of *ACSH*: Type II (λ_e in between 0.5 and $\sim$ 1.0)

step	state
0.	$[\ a^{10}, b^{10}, c^{10}]$
1.	$[\ a^{10}, b^{10}, c^{9}]$
	.....
41.	$[a^{2}, b^{7}, c^{5}]$
42.	$[[b^{2}]\ [b^{6}, c^{1}]]$
43.	$[[b^{4}]\ [b^{3}, c^{1}]]$
44.	$[[b^{4}]\ [b^{2}\ ][b^{2}, c^{1}]]$
45.	$[[b^{4}, c^{2}][b^{2}, c^{2}][b^{1}, c^{3}]]$
46.	$[[[b^{3}, c^{1}][b^{1}, c^{1}]][b^{2}, c^{2}][b^{1}, c^{3}]]$
47.	$[[[[b^{2}][a^{1}, b^{1}]][b^{1}, c^{1}]][a^{1}, b^{3}, c^{2}]]$
	$[a^{1}, b^{2}, c^{3}\ ]]$
140.	$[[[a^{1}, b^{1}][b^{1}, c^{1}]][[b^{2}][a^{1}, b^{1}]][b^{1}, c^{1}]]$
214.	$[[[[a^{16}, b^{3}, c^{2}][a^{5}, b^{5}, c^{1}][a^{3}, b^{2}]]$
	$[a^{4}, b^{3}]][[[[b^{2}][b^{2}][a^{1}, b^{1}]][a^{3}, b^{1}]]$
	$[a^{3}, b^{1}][b^{4}][a^{3}, b^{1}]]]$

Figure 6: An example of state transition of *ACSH*: Type III (λ_e in between 1.05 $\sim$ 2.33)

it is difficult to specifically increase the number of only membrane compounds then a cell is hardly to be divided (figure 7).

We performed the above simulations repeatedly and obtained the same results; four classes appeared according to λ_e. Because of the rules that were used in the above experiments were selected randomly at each rewriting steps, the results did not depend on specific rules. Therefore, λ_e can be a design parameter of the behavior of membranes in ACSH.

step	state
0.	$[a^{10}, b^{10}, c^{10}]$
1.	$[a^{12}, b^{5}, c^{9}]$
2.	$[a^{14}, b^{4}, c^{10}]$
3.	$[a^{13}, b^{6}, c^{11}]$
4.	$[a^{14}, b^{8}, c^{12}]$
5.	$[a^{9}, b^{9}, c^{10}]$
6.	$[a^{7}, b^{10}, c^{1}0]$
	.....
87.	$[a^{17}, b^{12}, c^{21}]$
	.....
147.	$[a^{14}, b^{20}, c^{29}]$
	.....
242	$[a^{17}, b^{50}, c^{44}]$
	.....
300.	$[a^{3}, b^{56}, c^{58}]$

Figure 7: An example of state transition of *ACSH*: Type IV (λ_e more than 2.5)

5 Discussion

P systems

Above two types of *ACS* correspond to a class of P systems which is a parallel molecular computing model proposed by G. Păun[6] and is based on the processing of multisets of objects in cell-like membrane structures. A P system is a multiset transformation system However, it is different from [2], since it includes a "*membrane*" in its computing mechanism. In the system *cells* are structured like living cells. The system itself is composed of several cells that are delimited from the neighboring cells, evolving internally. Technically, it is structured like a Venn diagram, each set in the diagram corresponding to a multiset. The sets under consideration are subsets of a unique set, which corresponds to a "*skin*" membrane, and is not allowed to intersect itself.

In the system, objects in each cell evolve with particular evolution rules. Any object, alone or together with one or more objects can evolve, can be transformed into other objects, can pass through one membrane, and can dissolve the membrane in which it is placed. All objects are active at the same time, in parallel; similarly, all membranes are active in parallel.

The computing power of P systems is equal to a Turing machine, proved in [6], and an algorithm to compute the SAT problem in linear time [7] was proposed. Various P systems have also been proposed and their mathematical properties have been investigated in [9, 10, 11].

ACS is an extension of P systems and a class of P systems. Although there have been proposed variants of P systems, many of them do not posses bio-chemical characteristics very much such as concentration of a chemical compounds, membrane compounds and so on. We have mathematical problems to investigate the correlation between ACS and P systems.

6 Conclusion

We proposed two types of membrane computing model that based on the Abstract Chemistries (ARMS). Throughout simulations we found that an artificial cell like chemoton and the cell cycle like behavior emerged. We discovered a parameter that describes the behavior of membranes. These models are closely related to P systems. Creating more concrete model of membrane computing and the investigation of the mathematical correlation between ACS and P systems are our future work.

Acknowledgment

We thank for fruitful discussion with Dr. S. Rasmussen, prof. Dr. W. Banzaf gave us some useful comments and encouraged us on the seminar at his university. Long Discussions with P. Dittrich were very insightful. And the authors would like to express many thanks to Dr. Gheorghe Păun for his useful comments, discussion, mathematical refinements. This research is supported by Grants-in Aid for Scientific Research No.11837005 from the Ministry of Education, Science and Culture in Japan.

References

[1] Fontana, W. and L.W. Buss, The arrival of the fittest: Toward a theory of biological organization. *Bulletin of Mathematical Biology* 56: 1–64, 1994.

[2] Berry, G. and G. Boudol, The chemical abstract machine. *Theoretical Computer Science* 96: 217–248, 1992.

[3] Field, R. J. and M. Burger, *Oscillations and Traveling Waves in Chemical Systems.* New York: John Wiley and Sons, 1985.

[4] Ganti, T, Organization of chemical reactions into dividing and metabolizing units: the chemotons. *Biosystems* 7: 189–195, 1975.

[5] Nicolis, G. and I. Prigogine, *Exploring Complexity, An Introduction.* San Francisco: Freeman and Company, 1989.

[6] Păun, G, Computing with Membranes, *Turku Center for Computer Science TUCS Technical Report No. 208* (submitted, also on http://www.tucs.fi), 1998.

[7] Păun, G, P Systems with Active Membranes: Attacking NP Complete Problems, *Center for Discrete Mathematics and Theoretical Computer Science CDMTCS-102* (also on http://www.cs.auckland.ac.nz/CDMTCS), 1999.

[8] Păun, G., G. Rozenberg, A. Salomaa, Membrane computing with external output, *Turku Center for Computer Science TUCS Technical Report No. 218* (submitted, on also http://www.tucs.fi), 1999.

[9] Păun, S. Yu, On synchronization in P systems, *CS Department TR No 539, Univ. of Western Ontario* (submitted, on also http://www.csd.uwo.ca/faculty/shu/TR539.html), 1999.

[10] Păun, T. Yokomori, Simulating H systems by P systems, submitted, 1999.

[11] Păun, T. Yokomori, Membrane computing based on splicing, submitted, 1999.

[12] Suzuki, Y. and H. Tanaka, Artificial proto-cell based on symbolic chemical system, submitted, 1999.

[13] Suzuki, Y. and H. Tanaka, Order parameter for a Symbolic Chemical System, *Artificial Life VI*:130-139, MIT press, 1998.

[14] Suzuki, Y. and H. Tanaka, Symbolic chemical system based on abstract rewriting system and its behavior pattern, *Journal of Artificial Life and Robotics*:1:211-219, Springer Verlag, 1997.

[15] Suzuki, Y. Tsumoto, S and H. Tanaka, Analysis of Cycles in Symbolic Chemical System based on Abstract Rewriting System on Multisets, *Artificial Life V*: 522-528. MIT press, 1996.