

Describing the Immune System Using Enhanced Mobile Membranes¹

Bogdan Aman²

*Romanian Academy, Institute of Computer Science
Blvd. Carol I no.8, 700505 Iași, Romania*

Gabriel Ciobanu³

*“A.I.Cuza” University of Iași, Faculty of Computer Science
Blvd. Carol I no.11, 700506 Iași, Romania*

Abstract

We investigate the enhanced mobile membranes, a new class of mobile membrane system in which new rules are introduced. The contextual evolution rules describe how an object from a membrane can evolve only in some context. The other rules describe the objective endocytosis and exocytosis. We use the class of enhanced mobile membrane system to model some evolutions in the immune system.

Keywords: immune system, membrane system, structural properties.

1 Introduction

Membrane systems (called also P systems) are introduced by Gh.Păun in [8,9] as a class of computing devices inspired by biology. The definition of this computing model starts from the observation that any biological system is a complex hierarchical structure with a flow of materials and information which underlies their functioning. The membrane computing deals with the evolution of systems composed by objects, rules and membranes nested in other membranes. The P systems with mobile membranes [7] describe the movement of membranes. The movement is given mainly by two operations: *exocytosis* and *endocytosis*, each of them working either in a “subjective” or an “objective” manner. According to [4], an “objective” movement is expressing that the moving membranes are controlled by objects

¹ Research partially supported by CEEX Grant 47/2005

² Email: baman@iit.tuiasi.ro

³ Email: gabriel@info.uaic.ro

placed inside the membranes that are passed through, and a “subjective” movement is expressing that a membrane controls its own moving.

The immune system is described in [6], a book which is revised every few years to keep the pace with the new discoveries in this field. The cells of the immune system work together with different proteins to seek out and destroy anything foreign or dangerous which enters our body. It takes some time for the immune cell to be activated, but once this happens there are very few hostile organisms having a chance. Immune cells are white blood cells produce in huge quantities in the bone marrow. There are a wide variety of immune cells, each of them with its own strength and weakness. Some seek out and engulf the invaders, while other destroy the infected or mutated body cells. Another type of cells, namely the B cells, have the ability to release special proteins called antibodies which mark intruders in order to be destroyed by macrophages. The immune system has also the ability to produce some cells able to remember enemies which it fought in the past. In this way, once the immune system recognizes an invader it attacks more quickly and strongly against it.

The structure of the paper is as follows. In Section 2 we give a short introduction to the immune system, whereas in Section 3 we present the enhanced mobile membranes, namely the mobile membranes presented in [7] with additional rules. Some examples are given in Section 4, where we express some mechanisms of the immune system using the new class of mobile membranes. Some structural properties of the mobile membranes are presented in Section 5. Conclusion and references end the paper.

2 Immune System

The immune system is one of the fascinating inventions of the nature which protects us against billions of bacteria, viruses, and other parasites. It is constantly on the alert, attacking at the first sign of an invasion, most of times without the knowledge of the host. The immune system is very complex, being composed of several types of cells and proteins which have different jobs to do in fighting foreign invaders.

Phagocytes are cells specialized in finding and “eating” bacteria, viruses, and dead or injured body cells. There are three main types of phagocytes:

- *Granulocytes* are the cells which take the first stand during an infection. They attack any invaders in large numbers, and “eat” them until they die. A small part of these kind of cells is specialized in attacking larger parasites such as worms.
- *Macrophages* (“big eaters”) have a slower response time than the granulocytes, but they are larger, live longer, and have far greater capacities. Macrophages also play a key part in alerting the rest of the immune system of invaders. Macrophages start out as white blood cells called monocytes. Monocytes which leave the blood stream turn into macrophages.
- *Dendritic cells* are “eater” cells and devour intruders, like the granulocytes and the macrophages. And like the macrophages, the dendritic cells help with the

activation of the rest of the immune system. They are also capable of filtering body fluids to clear them of foreign organisms and particles.

White blood cells called lymphocytes originate in the bone marrow but migrate to parts of the lymphatic system such as the lymph nodes, spleen, and thymus. There are two main types of lymphatic cells: T cells and B cells. The lymphatic system also involves a transportation system - lymph vessels - for transportation and storage of lymphocyte cells within the body. The lymphatic system feeds cells into the body and filters out dead cells and invading organisms such as bacteria. On the surface of each lymphatic cell are receptors that enable them to recognize foreign substances. These receptors are very specialized - each can match only one specific antigen. Once a macrophage phagocytizes a cell, it places portions of its proteins, called T cell epitopes, on the macrophage surface. These surface markers serve as an alarm to other immune cells which then infer the form of the invader. Macrophages engulf bacteria, process them internally, and display antigen parts on the cell surface together with MHC molecules. The MHC molecules (major histocompatibility complex) are molecules that display peptide antigen to T cells.

The only known function of dendritic cells is to present antigen to T cells, and the mature dendritic cells found in lymphoid tissues are by far the most potent stimulators of naive T cells. Immature dendritic cells persist in the peripheral tissues for variable lengths of time. When an infection occurs, they are stimulated to migrate via the lymphatics to the local lymphoid tissues, where they have a completely different phenotype. The dendritic cells in lymphoid tissue are no longer able to engulf antigens by phagocytosis or by macropinocytosis. However, they now express very high levels of long-lived MHC class I and MHC class II molecules; this fact enables them to stably present peptides from proteins acquired from the infecting pathogens. Tissue dendritic cells reaching the end of their life-span without having been activated by infection also travel via the lymphatics to local lymphoid tissue. Because they do not express the appropriate costimulatory molecules, these cells induce tolerance to any self antigens derived from peripheral tissues that they display.

The signals which activate tissue dendritic cells to migrate and mature after taking up antigen are clearly of key importance in determining whether an adaptive immune response will be initiated. For example, receptors which recognize lipopolysaccharide (LPS) are found on dendritic cells and macrophages, and these associate with the toll-like signalling receptor TLR4.

T cells undergo a selection process which retains those cells whose receptors interact effectively with various self peptide:self MHC ligands (positive selection), and removes those T cells that either cannot participate in such interactions (death by neglect) or recognize a self peptide:self MHC complex so well that they could damage host cells if allowed to mature; such cells are removed by clonal deletion (negative selection). Those T cells which mature and emerge into the periphery have therefore been selected for their ability to recognize self MHC:self peptide complexes without being fully activated by them. Cytotoxic or killer T cells (CD8+) do their work by releasing lymphotoxins, which cause cell lysis. Helper T cells (CD4+) serve

as regulators, and trigger various immune responses.

The B lymphocyte cell searches for antigen matching its receptors. If it finds such antigen, it connects to it, and inside the B cell a triggering signal is set off. Then a B cell needs proteins produced by helper T cells to become fully activated. When this happens, the B cell starts to divide to produce clones of itself. During this process, two new cell types are created: plasma cells and memory B cells.

Plasma cells are specialized in producing a specific protein, called an antibody, which responds to the same antigen which matched the B cell receptor. Antibodies are released from the plasma cell so that they can seek out intruders and help destroy them. Plasma cells produce antibodies at an amazing rate and can release tens of thousands of antibodies per second. When the Y-shaped antibody finds a matching antigen, it attaches to it. The attached antibodies serve as an appetizing coating for eater cells such as the macrophage. Antibodies also neutralize toxins and incapacitate viruses, preventing them from infecting new cells. Each branch of the Y-shaped antibody can bind to a different antigen, so while one branch binds to an antigen on one cell, the other branch could bind to another cell - in this way pathogens are gathered into larger groups which are easier for phagocyte cells to devour. Bacteria and other pathogens covered with antibodies are also more likely to be attacked by the proteins from the complement system. The complement system is one of the major mechanisms by which pathogen recognition is converted into an effective host defence against initial infection; it is composed of plasma proteins which lead to a cascade of reactions that occurs on the surface of pathogens and generates active components with various effector functions. This activity was said to 'complement' the antibacterial activity of antibody, hence the name.

Memory B Cells are the second cell type produced by the division of B cells. These cells form the basis of the immunological memory which ensures a more rapid and effective response on a second encounter with a pathogen and thereby provides lasting protective immunity. T cells can also produce memory cells with an even longer life span than memory B cells. Unlike memory T cells which can traffic to tissues owing to changes in cell-surface molecules that affect migration and homing, it is thought that memory B cells continue to recirculate through the same secondary lymphoid compartments which contain naive B cells, principally the follicles of spleen, lymph node, and Peyer's patch. Some memory B cells can also be found in marginal zones, though it is not clear whether these represent a distinct subset of memory B cells.

3 Mobile Membranes Systems

Let \mathbb{N} be a set of positive integers. Consider a finite alphabet Γ of symbols. A multiset over Γ is a mapping $u : \Gamma \rightarrow \mathbb{N}$. For any $a \in \Gamma$, the value $u(a)$ denotes the multiplicity of a in u (the number of occurrences of symbol a in u). Given two multisets u, v over Γ , for any $a \in \Gamma$ we have that $(u \uplus v)(a) = u(a) + v(a)$ is the multiset union, and $(u \setminus v)(a) = \max\{0, u(a) - v(a)\}$ is the multiset difference. We use the string representation of multisets used in the membrane systems. An example of a multiset is $u = aabca$ with $u(a) = 3$, $u(b) = 1$, $u(c) = 1$.

Starting from the mobile membranes presented in [7], we define a new class of mobile membranes, namely the enhanced mobile membrane systems. The distinction is made by three new rules ((b), (e) and (f)) which are inspired by some evolution of the immune system. The contextual evolution rule (b) states that a multiset from a membrane can evolve only in a certain context. The other two rules describe the objective endocytosis and exocytosis; rule (e) is called “forced endocytosis”, while (f) is called “forced exocytosis”.

Definition 3.1 A enhanced mobile membrane system is a construct

$$\amalg = (V, H, \mu, w_1, \dots, w_n, R), \quad \text{where:}$$

- (i) $n \geq 1$ (the initial *degree* of the system);
- (ii) V is an alphabet (its elements are called *objects*);
- (iii) H is a finite set of *labels* for membranes;
- (iv) μ is a membrane structure, consisting of n membranes, labelled with elements of H (two different membranes can have the same label). A membrane structure is a hierarchically arranged set of membranes, where we distinguish the external membrane (usually called the “skin” membrane) and several internal membranes; a membrane without any other membrane inside it is said to be elementary.
- (v) w_1, w_2, \dots, w_n are strings over V describing the *multisets of objects* placed in the n regions of μ ;
- (vi) R is a finite set of *developmental rules* of the following forms:
 - (a) $[u \rightarrow v]_m]_k$ for $k, m \in H, u, v \in V^*$; local evolution
 These rules are called *local* because the evolution of an multiset of objects u of membrane m is possible only when membrane m is inside membrane k . If the restriction of nested membranes is not imposed, that is, the evolution of the multiset of objects u in membrane m is allowed irrespective of where membrane m is placed, then we say that we have a global evolution rule, and write it simply as $[u \rightarrow v]_m$.
 - (b) $[[w]_m[u]_h]_k \rightarrow [[w]_m[v]_h]_k$ for $h, k, m \in H, u, v, w \in V^*$; contextual evolution
 These rules are called *contextual* because the evolution of an multiset of objects u of membrane h is possible only when membrane h is sibling with membrane m containing the multiset of objects w and both membranes h and m are placed inside the same membrane k . If the multiset of objects w is not specified, then the evolution is allowed only in the context of a sibling membrane m placed in the same membrane k .
 - (c) $[u]_h[]_m \rightarrow [[v]_h]_m$ for $h, m \in H; u, v \in V^*$; endocytosis
 An elementary membrane labelled h enters the adjacent membrane labelled m , under the control of the multiset of objects u . The labels h and m remain unchanged during this process; however the multiset of objects u may be modified to the multiset of objects v during the operation. Membrane m is not necessarily an elementary membrane.
 - (d) $[[u]_h]_m \rightarrow [v]_h[]_m$, for $h, m \in H; u, v \in V^*$;

exocytosis

An elementary membrane labelled h is sent out of a membrane labelled m , under the control of the multiset of objects u . The labels of the two membranes remain unchanged, but the multiset of objects u from membrane h may be modified during this operation. Membrane m is not necessarily elementary.

- (e) $[]_h[u]_m \rightarrow []_h[v]_m$ for $h, m \in H, u, v \in V^*$; forced endocytosis

An elementary membrane labelled h is engulfed into the adjacent membrane labelled m , under the control of the multiset of objects u . The labels h and m remain unchanged, but the multiset of objects u is transformed into the possibly new multiset of objects v during the operation. Membrane m is not necessarily elementary. The effect of this rule is similar to the effect of rule (c). The main difference from rule (c) is that the movement is not controlled by an object inside the moving membrane h , but by a multiset of objects u placed inside the membrane m which engulfs membrane h . Namely, the membrane which initiates the move is the passive membrane m , and not the active membrane h as in rule (c).

- (f) $[u[]_h]_m \rightarrow []_h[v]_m$ for $h, m \in H, u, v \in V^*$; forced exocytosis

An elementary membrane labelled h is sent out of a membrane labelled m under the control of the multiset of objects u . The labels of the two membranes remain unchanged, and the multiset of objects u is transformed into the possibly new multiset of objects v during the operation. Membrane m is not necessarily elementary. The effect of this rule is similar to the one of rule (d). The main difference from rule (d) is that the movement is not controlled by an object inside the moving membrane h , but by a multiset of objects u placed inside the membrane m which engulfs membrane h . Namely, the membrane which initiates the move is the passive membrane m , and not the active membrane h as in rule (d).

- (g) $[u]_h \rightarrow [v]_h[w]_h$, $h \in H, u, v, w \in V^*$; elementary division

In reaction with a multiset of objects u , the elementary membrane labelled h is divided into two membranes labelled h , with the multiset of objects u replaced in the two new membranes by possibly new multisets of objects v and w .

The rules of type (c) and (d) are used to simulate the movement of membranes in the membrane systems controlled by objects placed inside the moving membranes ("subjective" movement). The rules of type (e) and (f) are used to simulate the movement of membranes in the membrane system that are controlled by objects placed inside the membranes which are passed through ("objective" movement).

The contextual evolution rules, namely the rules of type (b), express the fact that an object from a membrane can evolve in the presence of an object placed in a sibling membrane, where the two sibling membranes are placed into the same membrane. The rules are called contextual because the evolution depends on the context.

In order to dissolve some membranes we use the special object $\delta \in V$ which, once is created by a rule of type (a) or (b), it dissolves the surrounding membrane.

The rules from the class of mobile membranes defined above are applied according to the following principles, some of them well-known in membrane computing:

- (i) In the rules of type (a) - (g) above, membranes m and k are called *passive*, while the membrane h is called *active*. For instance, a global evolution rule $[u \rightarrow v]_m$ of type (a) is considered to involve actively only the multiset of objects u , and not the membrane m which is passive. On the other hand, membrane k is active in the case of a local evolution rule.
- (ii) In each step we apply a maximal set of rules such that no further rule can be added to the set. The involved rules, objects and membranes are chosen nondeterministically, as follows: each object can participate in exactly one rule, while each membrane can participate as active in exactly one rule, or as passive in an arbitrary number of chosen rules. The application of the rules from the maximal set form an evolution step, called a transition step.
- (iii) When a membrane is moved across another membrane using a rule of type (c)-(f), its whole content (namely its objects) is moved.
- (iv) When a membrane is divided, then all its inner objects are replicated in the two new copies of the membrane. If at the moment when a membrane is divided there are objects which evolve by means of a rule of type (a) and (b), then the result of these rules are introduced in the new copies of the membrane.
- (v) If a membrane exits or is pushed through the skin membrane (by exocytosis), then its evolution stops even if there are rules of types (a), (b) or (g) which can be applied to it (provided that the membrane would be placed inside the skin membrane).
- (vi) Only elementary membranes can be moved by endocytosis and exocytosis through other membranes.
- (vii) The skin membrane can never be divided.

4 Examples

We consider some examples from the immune system, and illustrate how the resulting membrane systems evolves using the rules introduced in Section 3. The first example illustrates how the dendritic cells protect the human organism against infections.

Dendritic cells can engulf bacteria, viruses, and other cells. Once a dendritic cells engulfs a bacterium, it dissolves this bacterium and places portions of bacterium proteins on its surface. These surface markers serve as an alarm to other immune cells, namely helper T cells, which then infer the form of the invader. This mechanism makes sensitive the T cells to recognize the antigens or other foreign agents which triggers a reaction from the immune system. Antigens are often found on the surface of bacterium and viruses.

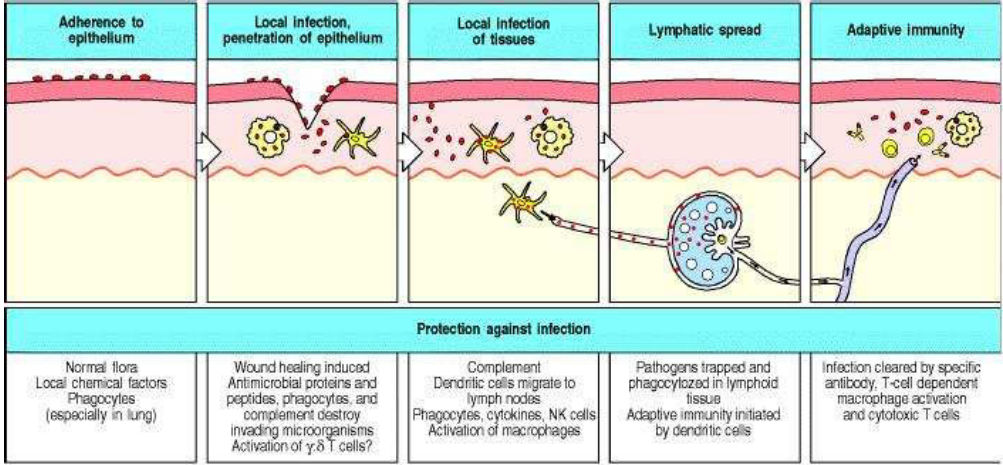


Figure 1. Protection against infection

In order to simulate the evolution presented in Figure 1, we need first to encode all the component of the immune system into a membrane system. This can be realized by associating a membrane to each component, and some objects to the signals, states and parts of molecules. For the steps done by the dendritic cells presented in Figure 1 we use the following encodings:

- dendritic cell - $[eat]_{DC}$

An immature dendritic cell is willing to eat any bacterium it encounters, so we translate it into a membrane labelled by DC which has inside an object eat used to engulf the bacterium. Once the dendritic cell matures, the object eat is consumed.

- bacterium cell - $[antigen]_{bacterium}$

A bacterium cell contains antigen so we simply represent it as a membrane labelled by $bacterium$ containing a single object $antigen$ which contains the information of the bacterium.

- lymph node - $[]_{lymph\ node}$

The lymph node is the place where the mature dendritic cells migrate in order to start the immune response, so we translate it into a membrane labelled by $lymph\ node$.

Using the above obtained membranes we can describe the membrane system as follows (here $skin$ stands for the body skin):

$$[[eat]_{DC} []_{lymph\ node}]_{skin} [antigen]_{bacterium}$$

with the following rules which describe its evolution:

- * $[antigen]_{bacterium} []_{skin} \rightarrow [[antigen]_{bacterium}]_{skin}$

A bacterium enters through the skin by performing an endocytosis rule in order to infect the body. The bacterium contains an object $antigen$ which represent its signature.

- * $[eat]_{DC} []_{bacterium} \rightarrow [eat []_{bacterium}]_{DC}$

Once an immature dendritic cell becomes sibling to a bacterium, it “eats” the bacterium by performing a forced endocytosis rule. Until this moment the bacterium has controlled its own movement; in this step of the evolution the movement becomes controlled by the dendritic cell which eats the bacterium.

$$* [[antigen]_{bacterium}]_{DC} \rightarrow [[antigen \ \delta]_{bacterium}]_{DC}$$

Once the bacterium has entered the dendritic cell, an object δ is created in order to dissolve the membrane *bacterium*, and the content of the bacterium is released into the dendritic cell.

$$* [antigen]_{DC} []_{lymph \ node} \rightarrow [[antigen]_{DC}]_{lymph \ node}$$

Once the dendritic cell contains parts of antigen, it enters the lymph node in order to activate a special class of T cells, namely the helper T cells.

$$* [[eat]_{DC}]_{lymph \ node} \rightarrow [[]_{DC}]_{lymph \ node}$$

Once the dendritic cell enters the lymph node it matures and the capacity to engulf bacteria disappear, namely the *eat* object is consumed.

Using only these few rules we can simulate the way a bacterium is engulfed and its content is displayed by the eater cell. The proteins produced by helper T cells activate the B cells.

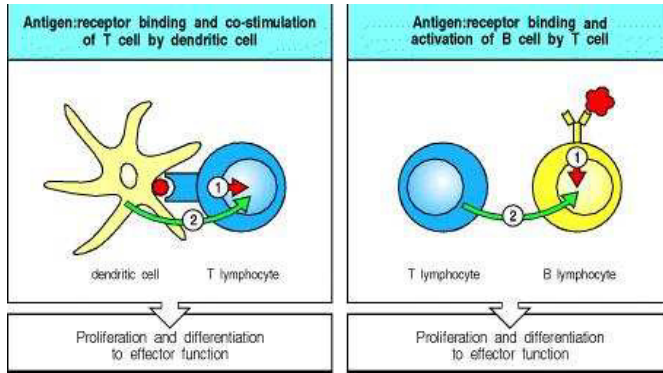


Figure 2. Activation of T cells and B cells

For the process of activating of helper T cell and B cell we use the following encodings:

- helper T cell - $[passive]_{helper \ T \ cell}$

A helper T cell is initially passive, so we represent it as a membrane labelled *helper T cell* in which the object *passive* is placed. When the cell is activated the object *passive* is transformed into *active*.

- B cell - $[passive]_{B \ cell}$

Such a B cell is initially passive, so we represent it as a membrane labelled *B cell* in which the object *passive* is placed. When the cell is activated the object *passive* is transformed into *active*.

The activation of the helper T cells and B cells is conditioned by the presence in the lymph node of the dendritic cells, and that is why we use the following contextual

evolution rules:

$$* \frac{[[antigen]_{DC}[passive]_{helper\ T\ cell}]_{lymph\ node}}{[[antigen]_{DC}[active]_{helper\ T\ cell}]_{lymph\ node}} \rightarrow$$

Once the dendritic cell containing parts of antigen enters the lymph node, it activates a special class of T cells, namely the helper T cells. This is denoted by changing the object *passive* to *active* in helper T cells.

$$* \frac{[[passive]_{B\ cell}[active]_{helper\ T\ cell}]_{lymph\ node}}{[[active]_{B\ cell}[active]_{helper\ T\ cell}]_{lymph\ node}} \rightarrow$$

Once the helper T cells are activated, the B cells that are sibling with them are the next cells which are activated.

The B cell searches for antigen matching its receptors. If it finds such antigen, then inside the B cell a triggering signal is set off. Using the proteins produced by helper T cells, the B cell starts to divide and produce clones of itself. During this process, two new cell types are created: plasma cells which produce an antibody, and memory cells which are used to “remember” specific intruders.

These examples motivate the introduction of the new class of mobile membranes; more exactly, they motivate the new rules and the way they can be used in modelling some biological systems.

5 Structural Properties in Mobile Membrane Systems

In this section we define a formal framework for the enhanced mobile membranes presented in Section 3. In this formal framework we study some properties, in particular the way the objects and membranes are arranged into the system, the depth of a membrane system, the evolution of the system in terms of its “top membranes”.

The order of neighbouring membrane structures and of objects placed at the same level in a large membrane structure is irrelevant; what matters is the relationships between membranes and objects. A rule of the form $a [b]_n \rightarrow b [a]_n$ has the same meaning as any of the rules $[b]_n a \rightarrow [a]_n b$, $a [b]_n \rightarrow [a]_n b$, and $[b]_n a \rightarrow b [a]_n$. We define the notion of structural congruence which clarifies this aspect, and reduces the number of rules written for a membrane system. We denote the membrane by M, M', M_i, N, N' , and the labels of the membranes by n, m, \dots

Definition 5.1 The set \mathcal{M} of membrane configurations M is defined by

$$M ::= w \mid [M]_n \mid M_1 M_2$$

where w describes multiset of objects placed inside membranes as in Definition 3.1, and by $M_1 M_2$ we mean the concatenation of two membrane configurations.

We can write w or $M_1 M_2$ omitting the surrounding membrane, because all the membrane structures are inside a *skin* membrane. There is a difference between the membrane configuration $M_1 M_2$ corresponding to sibling membranes and the membrane configuration $[M_1]_n M_2$ in which the membrane n containing the

configuration M_1 is sibling with the membrane M_2 .

Definition 5.2 The structural congruence \equiv_{mem} over \mathcal{M} is the smallest congruence relation satisfying the commutativity and associativity of the configuration concatenation: $M N \equiv_{mem} N M$, $M (N M') \equiv_{mem} (M N) M'$.

Proposition 5.3 *The structural congruence has the following properties:*

- $M \equiv_{mem} M$
- $M \equiv_{mem} N$ implies $N \equiv_{mem} M$,
- $M \equiv_{mem} N$ and $N \equiv_{mem} M'$ implies $M \equiv_{mem} M'$,
- $M \equiv_{mem} N$ implies $M M' \equiv_{mem} N M'$;
- $M \equiv_{mem} N$ implies $M' M \equiv_{mem} M' N$;
- $M \equiv_{mem} N$ implies $[M]_n \equiv_{mem} [N]_n$;

A “biological system” is represented by a mathematical model of a biological system; such a system evolves by passing from one configuration to another, and producing in this way a certain “behaviour”. An “observer” is placed outside the biological system, and watches its behaviour. In membrane systems the observer has the possibility of watching only the top-level membranes at any moment of the computation, where the set of top-level membranes TL is defined as follows:

- if $M = w$, then $TL(M) = \emptyset$
- if $M = [N]_n$, then $TL(M) = \{n\}$;
- if $M = M_1 M_2$, then $TL(M) = TL(M_1) \cup TL(M_2)$.

Denoting any rule $a), \dots, g)$ from our particular set of developmental rules by r , and by R a multiset of such rules, we use $M \xrightarrow{R} N$ to denote the transformation of a membrane system M into a membrane system N by applying a multiset of rules R . The *operational semantics* of the membrane systems is defined in terms of the relation \xrightarrow{R} by the following rules:

- (DRule) $M \xrightarrow{r} N$ for any developmental rule r
- (Empty) $M \xrightarrow{\emptyset} M$; (Amb) $\frac{M \xrightarrow{R} M'}{[M]_n \xrightarrow{R} [M']_n}$;
- (OutPar) $\frac{M \xrightarrow{R} M', N \xrightarrow{R'} N'}{M N \xrightarrow{R \uplus R'} M' N'}$; (InPar) $\frac{[M]_m \xrightarrow{R} [M']_m, [N]_m \xrightarrow{R'} [N']_m}{[M N]_m \xrightarrow{R \uplus R'} [M' N']_m}$;
- (Combine) $\frac{[M w]_n N_1 \xrightarrow{R} [M N'_1 u]_n, [M w]_n N_2 \xrightarrow{R'} [M N'_2 v]_n}{[M w]_n N_1 N_2 \xrightarrow{R \uplus R'} [M N'_1 N'_2 (((w \setminus u) \setminus v) \uplus (u \setminus w) \uplus (v \setminus w))]_n}$;
- (Break) $\frac{[M N_1 N_2 w]_n \xrightarrow{R} [M N_2 u]_n N'_1, [M N_1 N_2 w]_n \xrightarrow{R'} [M N_1 v]_n N'_2}{[M N_1 N_2 w]_n \xrightarrow{R \uplus R'} [M (((w \setminus u) \setminus v) \uplus (u \setminus w) \uplus (v \setminus w))]_n N'_1 N'_2}$;
- (Struc) $\frac{M \equiv_{mem} M', M' \xrightarrow{R} N', N' \equiv_{mem} N}{M \xrightarrow{R} N}$.
- (Step) $\frac{M \xrightarrow{R} N \text{ and } R \text{ is a maximal multiset of rules}}{M \Rightarrow_{mem} N}$;

In what follows we explain some of the above rules, and present examples for some of them in order to demonstrate how they are applied. The rule *(Empty)* states that if we do not apply a rule to a configuration, then the configuration remains the same. Example 5.4 illustrates how the rules *(Amb)*, *(OutPar)* and *(InPar)* work. Example 5.5 is an illustration of the rule *(Combine)*, while Example 5.6 is an illustration of the rule *(Break)*. Rule *(Step)* describes an evolution step called also a transition step.

Example 5.4 Let us consider the membrane system $[a\ a\ a\ [d\ d\ d\ d]_n]_m\ [\]_k$ and the rules $r_1 : [a\ a \rightarrow c]_m$ and $r_2 : [d\ d \rightarrow b]_n$. The evolution of this membrane system by applying a maximal set of rules, namely $r_1\ r_1\ r_2$ is:

$$\frac{(InPar) \frac{(InPar) \frac{[d\ d]_n \xrightarrow{r_2} [b]_n, [d\ d]_n \xrightarrow{r_2} [b]_n}{[d\ d\ d\ d]_n \xrightarrow{r_2\ r_2} [b\ b]_n} (Amb) \frac{[d\ d\ d\ d]_m \xrightarrow{r_2\ r_2} [b\ b]_m}{[d\ d\ d\ d]_m \xrightarrow{r_2\ r_2} [b\ b]_m} (InPar) \frac{[a\ a]_m \xrightarrow{r_1} [c]_m, [a\ a]_m \xrightarrow{\emptyset} [a]_m}{[a\ a\ a]_m \xrightarrow{r_1} [a\ c]_m}}{[a\ a\ a\ [d\ d\ d\ d]_n]_m \xrightarrow{r_1\ r_2\ r_2} [c\ a\ [b\ b]_n]_m} [\]_k \xrightarrow{\emptyset} [\]_k}{[a\ a\ a\ [d\ d\ d\ d]_n]_m \xrightarrow{r_1\ r_2\ r_2} [c\ a\ [b\ b]_n]_m\ [\]_k}$$

Example 5.5 Let us consider as a second example the membrane system from the Section 4, namely:

$$[\]_{skin} [antigen]_{bacterium} [antigen]_{bacterium}$$

and the evolution rule $r_1 : [\]_{skin} [antigen]_{bacterium} \rightarrow [[antigen]_{bacterium}]_{skin}$. The evolution of this membrane system by applying a maximal set of rules, namely $r_1\ r_1$ can be described as follows using the rule *(Combine)*:

$$\frac{[\]_{skin} [antigen]_{bacterium} \xrightarrow{r_1} [[antigen]_{bacterium}]_{skin}, [\]_{skin} [antigen]_{bacterium} \xrightarrow{r_1} [[antigen]_{bacterium}]_{skin}}{[\]_{skin} [antigen]_{bacterium} [antigen]_{bacterium} \xrightarrow{r_1\ r_1} [[antigen]_{bacterium} [antigen]_{bacterium}]_{skin}}$$

Example 5.6 Let us consider as a third example the membrane system:

$$[[antigen]_{bacterium} [antigen]_{bacterium}]_{skin}$$

and the evolution rule $r_1 : [[antigen]_{bacterium}]_{skin} \rightarrow [\]_{skin} [antigen]_{bacterium}$. The evolution of this membrane system by applying a maximal set of rules, namely $r_1\ r_1$ can be described as follows using the rule *(Break)*:

$$\frac{[[antigen]_{bacterium}]_{skin} \xrightarrow{r_1} [\]_{skin} [antigen]_{bacterium}, [[antigen]_{bacterium}]_{skin} \xrightarrow{r_1} [\]_{skin} [antigen]_{bacterium}}{[[antigen]_{bacterium} [antigen]_{bacterium}]_{skin} \xrightarrow{r_1\ r_1} [\]_{skin} [antigen]_{bacterium} [antigen]_{bacterium}}$$

We introduce a barbed bisimulation which equates systems if they are indistinguishable under certain observations. The key ingredient of the barbed bisimulation is the notion of barb. A barb predicate describes the observed elements of a certain structure.

Definition 5.7 A barb \downarrow^{mem} is defined inductively by the following rules:

$$\begin{aligned} M \downarrow^{mem} n & \quad \text{if } n \in TL(M) \\ M_1 \cdots M_k \downarrow^{mem} n_1 \cdots n_k & \quad \text{if } n_i \in TL(M_i) \text{ for each } i \end{aligned}$$

We write $M \Downarrow^{mem} n$ if either $M \downarrow^{mem} n$ or $M \Rightarrow_{mem}^+ M'$ and $M' \downarrow^{mem} n$, where \Rightarrow_{mem}^+ is the transitive closure of \Rightarrow_{mem} .

Formally, we have:

$$\begin{aligned}
M \downarrow^{mem} n &\stackrel{def}{=} M \equiv_{mem} [M_1]_n, M_2. \\
M \Downarrow^{mem} n &\stackrel{def}{=} \text{either } M \downarrow^{mem} n \text{ or } M \Rightarrow_{mem}^+ M' \text{ and } M' \downarrow^{mem} n.
\end{aligned}$$

The following result reflects a relationship between structural congruence and barbs predicates.

Proposition 5.8 *Structurally congruent membrane structures have the same top level membranes. If $M \equiv_{mem} N$, then $M \downarrow^{mem} n$ iff $N \downarrow^{mem} n$, for all $n \in TL(M, N)$.*

The set of membrane labels ML is defined as follows:

$$\begin{aligned}
&\text{if } M = w, \text{ then } ML(M) = \emptyset \\
&\text{if } M = [N]_n, \text{ then } ML(M) = ML(N) \cup \{n\}; \\
&\text{if } M = M_1 M_2, \text{ then } ML(M) = ML(M_1) \cup ML(M_2).
\end{aligned}$$

If a system contains a top level membrane after applying a number of computation steps, then a structurally congruent membrane system contains the same top level membrane after applying the same number of computation steps.

Proposition 5.9 *If $M \equiv_{mem} N$, then $M \Downarrow^{mem} n$ iff $N \Downarrow^{mem} n$ for all $n \in ML(M, N)$.*

6 Conclusion

The paper presents a new class of membrane systems with mobile membranes, namely the enhanced mobile membranes: besides a “subjective” form of movement, inspired by endocytosis / exocytosis, also a form of “objective” movement is introduced. In the subjective movement rules, the execution of a rule causing the movement of a membrane is triggered by an object residing inside the membrane; on the other hand, in objective movement rules the execution of the rule is triggered by an object residing in a sibling or in a parent membrane of the membrane which is moved. We motivate the need for these new rules by modelling some aspects of the immune system, where cells “eat” other cells (so the membrane movement is controlled by the eater, not by the eaten membrane). Finally we define some formal properties of the enhanced mobile membranes.

The mobile membranes are related to mobile ambients [4], having certain similarities and some common concepts. Ambients can also be nested, having a hierarchical structure. An ambients has a name, and can move with all the computations and subambients it contains. However the mobile membrane systems has local rules, very useful in modelling different behaviours of the system components. A formal relationship is given in [1] where a translation from mobile ambients to mobile membranes of [7] is defined; the soundness of this translation is provided by an operational correspondence result. Both formalisms can be used to model biological phenomena. In [5] we describe the sodium-potassium exchange pump using both mobile membrane systems and mobile ambients. A description of the sodium-potassium pump is given in [3] by using simply P systems (membrane systems

without mobility).

It is possible to define various quantitative concepts in enhanced mobile membranes. Based on the relationship between ambients and mobile membranes, and on the study of some quantitative properties of mobile ambients in [2], we intend to introduce domains, timers and capacities in enhanced mobile membranes.

References

- [1] B. Aman, G. Ciobanu. Translating Mobile Ambients into P Systems. *Electronic Notes in Theoretical Computer Science* vol.171, Elsevier, 11-23, 2007.
- [2] B. Aman, G. Ciobanu. Timers and Proximities for Mobile Ambients. *Proceedings CSR 2007*, Lecture Notes in Computer Science vol.4649, Springer, 33-43, 2007.
- [3] D. Besozzi, G. Ciobanu. A P System Description of the Sodium-Potassium Pump. *Workshop on Membrane Computing*, Lecture Notes in Computer Science vol.3365, Springer, 211-223, 2005.
- [4] L. Cardelli, A. Gordon. Mobile Ambients. *Theoretical Aspects of Computer Software*, Lecture Notes in Computer Science vol.1378, Springer, 140-155, 1998.
- [5] G. Ciobanu, B. Aman. On the Relationship Between Membranes and Ambients. *BioSystems*, Elsevier, 2007.
- [6] C.A. Janeway, P. Travers, M. Walport, M.J. Shlomchik. *Immunobiology - The immune system in health and disease*, 5th Edition, Garland Publishing, 2001.
- [7] S.N. Krishna, Gh. Păun. P Systems with Mobile Membranes. *Natural Computing* vol.4, 255-274, 2005.
- [8] Gh. Păun. Computing with membranes. *Journal of Computer and System Sciences* vol.61, 108-143, 2000.
- [9] Gh. Păun. *Membrane Computing. An Introduction*. Springer, 2002.