# Simulating Membrane Systems and Dissolution in a Typed Chemical Calculus

Bogdan Aman[1], Péter Battyányi[2], Gabriel Ciobanu[1], and György Vaszil[2]

[1] Romanian Academy, Institute of Computer Science
Blvd. Carol I no.8, 700505 Iaşi, Romania
baman@iit.tuiasi.ro
gabriel@info.uaic.ro

[2] Department of Computer Science, Faculty of Informatics
University of Debrecen
Kassai út 26, 4028 Debrecen, Hungary
battyanyi.peter@inf.unideb.hu
vaszil.gyorgy@inf.unideb.hu

**Summary.** We present a transformation of membrane systems, possibly with promoter/inhibitor rules, priority relations, and membrane dissolution, into formulas of the chemical calculus such that terminating computations of membranes correspond to terminating reduction sequences of formulas and vice versa. In the end, the same result can be extracted from the underlying computation of the membrane system as from the reduction sequence of the chemical term. The simulation takes place in a typed chemical calculus, but we also give a short account of the untyped case.

## 1 Introduction

In the present paper we continue the investigations started in [2, 3] concerning the possibility of defining the semantics of membrane systems with rewriting logic [1, 2] in order to obtain a logical description of membrane system computations.

The direct precedent of our work is [7] where a logical description of simple membrane systems was given using the $\gamma$-calculus of Banâtre and Le Métayer from [6] (see also [4] for more details). Their aim was to free the expression of algorithms from the sequentiality which is not inherently present in the problem to be solved, that is, the sequentiality which is implied by the structure of the computational model on which the given algorithm is to be performed. They called their calculus chemical calculus, and the underlying computational paradigm the chemical paradigm of computation while the execution model behind them closely resembles the way chemical reactions take place in chemical solutions. A chemical "machine" can be thought of as a symbolic chemical solution where data can be seen as molecules and operations as chemical reactions. If some molecules satisfy a reaction condition, they are replaced by the result of the reaction. If no reaction is possible, the program terminates. Chemical solutions are represented by multisets. Molecules interact freely according to reaction rules which results in an implicitly parallel, non-deterministic, distributed model.

In what follows, using a slightly modified variant of the operational semantics of membrane systems presented in [3], we show how to transform a membrane system with rules using promoters/inhibitors (see [8]), priorities, and also the possibility of membrane dissolution (introduced already in [9]), into formulas of the chemical calculus, such that terminating computations of the membrane system correspond to terminating reduction sequences of formulas and vice versa.

## 2 Preliminaries

In this section we present the basic notions and notations we are going to use. For a comprehensive treatment of membrane systems ranging from the basic definitions to their computational power, see the the monographs [10, 11], for more information on the chemical calculus, we refer to [4, 5].

A finite multiset over an alphabet $V$ is a mapping $m : V \to \mathbb{N}$ where $\mathbb{N}$ denotes the set of non-negative integers, and $m(a)$ for $a \in V$ is said to be the multiplicity of $a$ in $V$. We say that $m_1 \subseteq m_2$ if for all $a \in V$, $m_1(a) \leq m_2(a)$. The union or sum of two multisets over $V$ is defined as $(m_1 + m_2)(a) = m_1(a) + m_2(a)$, the difference is defined for $m_2 \subseteq m_1$ as $(m_1 - m_2)(a) = m_1(a) - m_2(a)$ for all $a \in V$. The multiset $m$ can also be represented by any permutation of a string $w = a_1^{m(a_1)} a_2^{m(a_2)} \dots a_n^{m(a_n)} \in V^*$, where if $m(x) \neq 0$, then there exists $j$, $1 \leq j \leq n$, such that $x = a_j$. The set of all finite multisets over an alphabet $V$ is denoted by $\mathcal{M}(V)$, the empty multiset is denoted by $\emptyset$ as in the case of the empty set.

### 2.1 Membrane systems

A membrane system, or P system is a structure of hierarchically embedded membranes, each having a label and enclosing a region containing a multiset of objects and possibly other membranes. The unique out-most membrane is called the skin membrane. The membrane structure is denoted by a sequence of matching parentheses where the matching pairs have the same label as the membranes they represent. We assume the membranes are labelled by natural numbers $\{1, \dots, n\}$, and we use the notation $m_i$ for the membrane with label $i$. Each membrane $m_i$, except for the skin membrane, has its parent membrane, which we denote by $\mu(m_i)$. As an abuse of notation $\mu$ stands for both the membrane structure and both for the function determining the parent membrane of a membrane. To facilitate presentation we assume that $\mu(m_j) = m_i$ implies $i < j$.

The evolution of the contents of the regions of a P system is described by rules associated to the regions. The system performs a computation by passing from one configuration to another one, applying the rules synchronously in each region. In the variant we consider in this paper, the rules are multiset rewriting rules given in the form of $u \to v$ where $u, v$ are multisets, and they are applied in the maximal parallel manner, that is, as many rules are applied in each region as possible. The end of the computation is defined by the following halting condition: A P system halts when no more rules can be applied in any of the regions; the result is a number, the number of objects in a membrane labelled as output.

A *P system* of degree $n \geq 1$ is a construct

$$\Pi = (O, \mu, w_1, \dots, w_n, R_1, \dots, R_n, \rho_1, \dots, \rho_n)$$

where

- $O$ is an alphabet of objects,
- $\mu$ is a membrane structure of $n$ membranes,
- $w_i \in \mathcal{M}(O),\ 1 \leq i \leq n$, are the initial contents of the $n$ regions,
- $R_i,\ 1 \leq i \leq n$, are the sets of evolution rules associated to the regions; they are of the form $u \to v$ where $u \in \mathcal{M}(O)$ and $v \in \mathcal{M}(O \times tar)$ where $tar = \{here, out\} \cup \{in_j \mid 1 \leq j \leq n\}$, and
- $\rho_1, \ldots, \rho_n$ are the priority rules associated with membranes $m_1, \ldots, m_n$.

The evolution rules of the system are applied in the non-deterministic, maximally parallel manner to the $n$-tuple of multisets of objects constituting the configuration of the system. A configuration is the sequence $C = (v_1, \ldots v_n, \mu_C)$ where $v_i \in O^*,\ 1 \leq i \leq n$ are the contents of the membranes, and $\mu_C$ is the current membrane structure. For two configurations $C_1 = (u_1, \ldots, u_n, \mu_{C_1})$ and $C_2 = (v_1, \ldots, v_n, \mu_{C_2})$, we can obtain $C_2$ from $C_1$, denoted as $C_1 \Rightarrow C_2$, by applying the rules of $R_1, \ldots, R_n$. Let $\mathcal{R} = R_1 \cup R_2 \cup \cdots \cup R_n$, where $R_i = \{r_{i1}, \ldots, r_{ik_i}\}$ is the set of rules corresponding to membrane $m_i$. The application of $u \to v \in R_i$ in the region $i$ means to remove the objects of $u$ from $u_i$ and add the new objects specified by $v$ to the system. The rule application in each region takes place in a non-deterministic and maximally parallel manner. This means that the rule application phase finishes, if no rule can be applied anymore in either region. As a result, each region where rule application took place, is possibly supplied with elements of the set $O \times tar$. We call a configuration which is a multiset over $O \cup O \times tar$ an intermediate configuration. If we want to emphasize that $C = (w_1, \ldots, w_n, \mu)$ consists of multisets over $O$, we say that $C$ is a proper configuration. Rule applications can be preceded by priority check, if priority relations are present. Let $\rho_i \subseteq R_i \times R_i$ $1 \leq i \leq n$ be the (possibly empty) priority relations. Then $r \in R_i$ is applicable only if no $r' \in R_i$ can be applied with $(r', r) \in \rho_i$. We may also denote the relation $(r', r) \in \rho_i$ by $r' > r$.

In the next phase the objects coming from $v$ should be added to the regions as specified by the target indicators associated to them. If $v$ contains a pair $(a, here) \in O \times tar$, then $a$ is placed in region $i$, the region where the rule is applied. If $v$ contains $(a, out) \in O \times tar$, then $a$ is added to the contents of the parent region of region $i$; if $v$ contains $(a, in_j) \in O \times tar$ for some region $j$ which is contained inside the region $i$ (so region $i$ is the parent region of region $j$), then $a$ is added to the contents of region $j$.

The symbol $\delta$ marks a region for dissolution. When it is introduced in the membrane by a rule, after having finished the maximal parallel and communication steps, the actual membrane disappears. Its objects move to the parent membrane and its rules can not be applied anymore.

We can render promoter/inhibitor sets, *prom*/*inhib*, to each rule $r = (u \to v) \in \mathcal{R}_i$. The promoter/inhibitor sets belonging to $r$ are subsets of $O$. When $r$ is going to be applied they act as follows: $r$ can be applied to the content $w_i$ of membrane $m_i$ only if every element of *prom* is present in $w$ and no element of *inhib* can be found in $w$.

### 2.2 The chemical calculus

We give a brief summary of the chemical calculus following the presentation in [4] and [5]. Chemical programming is the formal equivalent of Gamma programming, which is a higher order multiset manipulating program language. Like Gamma programming, the chemical calculus is also based on the chemical metaphor: data are represented by $\gamma$-terms, which are called molecules, and reactions between them are represented by rewrite rules. We begin with the basic definitions. The syntactical elements of molecules, reaction conditions, and patterns, denoted by $M$, $C$ and $P$, respectively, are defined as follows.

$$M := x \mid (M_1, M_2) \mid \langle M \rangle \mid \gamma(P)[C].M$$

where $x$ is a variable standing for any molecule, $(M_1, M_2)$ is a compound molecule built with the commutative and associative "," constructor operator, $\langle M \rangle$ is called a solution, and $\gamma(P)[C].M$ is called a $\gamma$-abstraction with pattern $P$, reaction condition $C$, result $M$. The $\gamma$-abstraction encodes a rewriting rule: when the pattern $P$ is respected and the condition $C$ is met, a substituted variant of $M$ is created as a result. A pattern is

$$P := x \mid (P_1, P_2) \mid \langle P \rangle,$$

where $x$ matches any molecule, $(P_1, P_2)$ matches a compound molecule, and $\langle P \rangle$ matches an inert solution, that is, a solution where no reaction can occur: it consists entirely of solutions or entirely of $\gamma$-abstractions. (The contained solutions can still be active, however.)

The solution $\langle M \rangle$ encapsulates the molecule $M$ which is inside the solution, and thus, insulated from molecules outside the solution. The contents of solutions can only be changed by reactions which occur inside the solution.

Now we define how patterns are matched, which requires the notion of substitution. A *substitution* is a mapping $\phi$ from the set of variables to the set of molecules. We can define the application of a substitution to as follows:

$$\begin{aligned} \phi x &= \phi(x) \\ \phi(M_1, M_2) &= \phi M_1, \phi M_2 \\ \phi\langle M \rangle &= \langle \phi M \rangle \\ \phi(\gamma(P)[C].M) &= \gamma(P)[C].\phi' M, \end{aligned}$$

where $\phi'$ is obtained from $\phi$ by removing from the domain all the variables which occur in $P$.

The result of a match is an assignment of molecules to variables. The first argument of *match* is a pattern, the second one is a molecule, its value is a substitution. Let $x$ denote a variable, $P$ a pattern, and $M$ a molecule. Then we define

$$\begin{aligned} match(x, M) &= \{x \mapsto M\} \\ match((P_1, P_2), (M_1, M_2)) &= match(P_1, M_1) \circ match(P_2, M_2) \\ match(\langle P \rangle, \langle M \rangle) &= match(P, M) \text{ provided } inert(M) \\ match(P, M) &= \boldsymbol{fail} \text{ in every other case,} \end{aligned}$$

where $\circ$ denotes the operation of function composition.

The reaction rule is defined as

$$\gamma(P)[C].M, N \to \phi M,$$

where $match(P, N) = \phi$ assigns values to variables in such a way that $\phi(C)$ holds in the typed case or reduces to *true* in the untyped case. In this case *true* can be a special constant defined in advance, for example, $true \rightleftharpoons \gamma\langle x \rangle[x].x$.

We can define an operator *replace* (cf. [5]) which does not vanish in the course of the reduction:

$$replace\ P\ by\ M\ if\ C \rightleftharpoons let\ rec\ f = \gamma(P)[C].M, f\ in\ f.$$

Then the new operator obeys the following reduction rule:

$$replace\ P\ by\ M\ if\ C, N \to replace\ P\ by\ M\ if\ C, \phi(M),$$

where $match(P, N) = \phi$ and either $\phi(C)$ is true or it reduces to $true$.

At this point we should mention that the simulation takes place in the typed $\gamma$-calculus ([5]), because it is more convenient to talk about equality and comparison of integer values, than to check whether the conditional part of an untyped $\gamma$-expression reduces to *true* (which is, in fact, undecidable in the general case). We could, however, restrict the $\gamma$-expressions taking part in the simulation in such a way that their conditional parts form a fragment of the $\gamma$-calculus that is decidable with respect to equality. (We can take, e. g., the $\gamma$-calculus equivalents of Church numerals and define Boolean operations on them.)

## 3 Results

First we introduce molecules for the description of membrane system configurations.

**Notation 1** *Let* $[x, y] = (\langle x \rangle, y)$, *and* $[x_1, \ldots, x_n, x_{n+1}] = [[x_1, \ldots, x_n], x_{n+1}]$.

*Remark 1.* Let $P = [x_1, x_2, \ldots, x_l]$ be a pattern in the sense of the previous section, and $M = [s_1, s_2, \ldots, s_l]$, where $s_1, \ldots, s_l$ are arithmetical expressions, i.e. expressions composed of natural numbers, variables and arithmetical operations. If we assume that none of the $x_i$ appears among the free variables of $s_1, \ldots, s_l$, then $match(P, M) = \Phi \neq fail$ implies $\Phi = [x_1/s_1, x_2/s_2, \ldots, x_l/s_l]$, where $\Phi$ is the simultaneous substitution formed by the substitutions $[x_1/s_1], \ldots, [x_l/s_l]$. In other words, in this special case, the molecule $[x_1, x_2, \ldots, x_l]$ behaves as an ordered tuple.

If we use $a$, $b$ as variables for elements of $O$ and $r$ as a rule variable, respectively, then we say that a rule $r = u \to v \in R_i$ is valid with respect to the configuration $(w_1, \ldots, w_n, \mu)$ if the following conditions hold:

1. membrane structure $\mu$ contains membrane $m_i$,
2. $(\forall a \in prom_r)\ (w_i(a) \geq 1)$,
3. $(\forall a \in inhib_r)\ (w_i(a) = 0)$, and
4. $(\forall a \in O)(\forall\ 1 \leq j \leq n)\ (v(a, in_j) \geq 1)$ implies that $\mu$ contains the membrane $m_j$ ($m_j$ is not dissolved) and $\mu(m_j) = m_i$, namely $m_i$ is the parent membrane of $m_j$.

where $prom_r \subseteq O$ and $inhib_r \subseteq O$ denotes the set of promoters and inhibitors associated to rule $r$, respectively.

A description of a membrane system configuration as above is a molecule of the form

$$\begin{aligned}
Descr = [&c_{11}, \ldots, c_{1k}, \ldots, c_{n1}, \ldots, c_{nk},\\
&\overline{c}_{11}, \ldots, \overline{c}_{1k}, \ldots, \overline{c}_{n1}, \ldots, \overline{c}_{nk},\\
&d_1, \ldots, d_n,\\
&p_{11}, \ldots, p_{1k_1}, \ldots, p_{n1}, \ldots, p_{nk_n}],
\end{aligned}$$

where $c_{ij}$ and $\overline{c}_{ij}$ are natural numbers $(1 \leq i \leq n, 1 \leq j \leq k)$, $d_i \in \{0,1\}$ $(1 \leq i \leq n)$ and $p_{ik_j} \in \{0,1\}$ $(1 \leq i, j \leq n)$. If $N$ is a description we denote by $c_{ij}$, $\overline{c}_{ij}$, etc. the respective parts of $N$.

Let $C = (\mu, w_1, \ldots, w_n)$ be an (intermediate) configuration. A description $Descr(C)$ corresponding to $C$ is a description, where $c_{ij} = w_i(a_j)$ and $\overline{c}_{ij} = w_i(a_j, here) + \sum_{p \neq i, \mu(m_i) = m_p} w_p(a_j, in_i) + \sum_{\mu(p)=i} w_p(a_j, out)$ with $(1 \leq i, p \leq n)$ and $(1 \leq j \leq k)$. Here $\mu(p)$ denotes the parent membrane of $m_p$, and recall that $w(a)$ denotes the number of elements $a$ in the multiset $w$. Intuitively, $c_{ij}$ stands for the number of occurrences of $a_j$ in $m_i$, and $\overline{c}_{ij}$ denotes the location of the targeted elements of $O$. Moreover, $d_i = 1$ iff $m_i$ is dissolved or under dissolution and $p_{ik_j}$ describes the validity of rules: rule $r_{ik_j}$ is valid iff $p_{ik_j} = 1$, if $C$ is a proper configuration. If $C \to^* C'$, and $C'$ is an intermediate configuration and there are no proper configurations in the reduction sequence other than $C$, then $p'_{ik_j} = 1$ in the description of $C'$ iff $p_{ik_j} = 1$ in the description of $C$. Observe that if $C$ is a proper configuration then $\overline{c}_{ij} = 0$ for every possible $i$ and $j$. When a configuration is proper, $d_i = 1$ implies $w_i = 0$.

A pattern for a description is a tuple of the form

$$\begin{aligned}
S = [&x_{m_1 a_1}, \ldots, x_{m_1 a_k}, \ldots, x_{m_n a_1}, \ldots, x_{m_n a_k},\\
&\overline{x}_{m_1 a_1}, \ldots, \overline{x}_{m_1 a_k}, \ldots, \overline{x}_{m_n a_1}, \ldots, \overline{x}_{m_n a_k},\\
&x_{d_1}, \ldots, x_{d_n}, x_{r_{1k_1}}, \ldots, x_{r_{nk_n}}].
\end{aligned} \tag{1}$$

Let $\Pi = (O, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n, \rho_1, \ldots, \rho_n)$ be a P system, and let $C' = (w'_{k_1}, \ldots, w'_{k_j}, \mu')$ be a proper configuration obtained from the initial configuration in a finite number of computational steps, where $1 \leq k_1 < \ldots < k_j \leq n$. Then the description of $C'$ relative to $\mu$ is the description obtained from $Descr(C')$ when we set $d_i = 1$ for $i \notin \{k_1, \ldots, k_j\}$ and $c_{ij} = 0$ $(1 \leq j \leq k)$ and $p_{il} = 0$ for every rule $r_{il} \in R_i$. That is, we supplement $Descr(C')$ as if it were a description of an $n$-ary membrane system by treating the missing membranes as empty membranes. We denote the description of a configuration $C'$ relative to $\mu$ by $Descr_\mu(C')$.

Because a description should also contain information about the structure of the original P system itself, we append a representation of the function $\mu$ at the end of each description. Let $\Pi$ be a P system of order $n$ as before. Then a tuple $[p_2, \ldots, p_n]$ of length $n-1$ is appended to every description in the simulation with the following meaning: if membrane $m_j$ has membrane $m_i$ as its parent, then $p_j = i$. Since the $Skin$ has no parent membrane, numbering begins with 2. Likewise, a description pattern is expanded with the tuple $[x_{p_2}, \ldots, x_{p_n}]$. Since the structure of the original P system remains the same in the course of the simulation process, we do not indicate the appended values for $\mu$, they are implicitly understood to be there.

With this in hand we are able to define the molecule in charge for deciding rule validity. Let $r = u \to v \in R_i$, and $S$ be a description pattern . Then let

$$\begin{aligned} Cond(r) = (x_{d_i} = 0 \wedge & \\ & \bigwedge_{1\leq j\leq k} (a_j \in prom_r \supset x_{m_i a_j} \geq 1) \wedge \\ & \bigwedge_{1\leq j\leq k} (a_j \in inhib_r \supset x_{m_i a_j} = 0)\big) \wedge \\ & \bigwedge_{1\leq l\leq k} \bigwedge_{1\leq j\leq n} \big(v(a_l, in_j) \geq 1 \supset x_{d_j} = 0 \wedge \\ & (\bigvee_{l_0=i>l_1>\ldots>l_{s-1}>j=l_s} (\bigwedge_{1\leq t\leq s} x_{p_t} = l_{t-1} \wedge \bigwedge_{1\leq q\leq s-1} x_{d_q} = 1))\big) \end{aligned} \tag{2}$$

The last row expresses the fact that either $m_i$ is the parent of $m_j$, or $m_i$ is an ancestor of $m_j$ and all the intermediate parent membranes have been dissolved in the construction.

Now rule validity can be expressed as

$$Val(r) = replace\ [S, 0]\ by\ [S[x_r/1], 0]\ if\ Cond(r)$$

where the value 0 plays a role of synchronization to be specified later on. We remark that if a rule $r$ is determined to be valid in this phase of the simulation, then $r$ remains valid in the course of the simulation of a maximal parallel step.

**Discussion 1** *At this point, we can also incorporate in the simulation of a membrane system the priority rules, if present. Let $(\rho_1, \ldots, \rho_n)$ be the tuple prescribing the priority relations in the membranes of the given P system. We define molecules determining the validity of rules when priority is present. Assume $r \in R_i$. We distinguish two cases:*

- *There does not exist $r' \in R_i$ such that $r' > r$. Then $Val_\rho(r)$ is defined as $Val(r)$ above.*
- *There are rules $r_1, \ldots, r_j \in R_i$ such that $r_l > r$ $(1 \leq l \leq j)$. Let $S$ be a description pattern and denote by $Cond(r)$ the conditional part of $Val(r)$ defined in Equation (2). Then*

$$\begin{aligned} Val_\rho(r) = \big(& replace\ [S, 0]\ by\ [S[x_r/1], 0] \\ & if\ (Cond(r) \wedge \bigwedge_{1\leq l\leq j} x_{r_l} = 0), \\ & replace\ [S, 0]\ by\ [S[x_r/0], 0] \\ & if\ (x_r = 1 \wedge (\bigvee_{1\leq l\leq j} x_{r_l} = 1)\big). \end{aligned}$$

Now we can turn to the main part of the simulation. The conditions of rule application must reflect now the fact that the rule is executable together with the conditions that make it valid.

**Definition 1.** *Let $r = u \to v \in R_i$, and let $S$ be a description pattern. Then the molecule describing the effect of an execution of $r$ is defined as*

$$\begin{aligned} App(r) = replace\ [S, 1]\ by\ [\,apply(S, r), 1]\ if & \\ \big(x_r = 1 \wedge \bigwedge_{1\leq j\leq k} (u(a_j) \leq x_{m_i, a_j}), & \end{aligned}$$

*where*

$$apply(S,r)(x_{m_s a_t}) = \begin{cases} x_{m_s a_t} - u(a_t) & \text{if } s = i, \\ x_{m_s a_t} & \text{otherwise }, \end{cases}$$

$$apply(S,r)(\overline{x}_{m_s a_t}) = \begin{cases} \overline{x}_{m_s a_t} + v(a_t, here) & \text{if } s = i, \\ \overline{x}_{m_s a_t} + v(a_t, in_j) & \text{if } s = j \neq i, \\ \overline{x}_{m_s a_t} + v(a_t, out) & \text{if } s = \mu(i), \end{cases}$$

$$apply(S,r)(x_{d_j}) = \begin{cases} 1 & \text{if } v(\delta) = 1, \\ x_{d_j} & \text{otherwise }, \end{cases}$$

$$apply(S,r)(x_r) = x_r.$$

Here we made use of the implicit stipulation that $S$ is of the form as in Equation (1), which is indeed the case if we ignore variable renaming.

The next group of rules is the set of communication rules. In what follows, we define the chemical calculus equivalents of communication steps.

**Definition 2.**

$$Msg = replace\ [S, 2]\ by\ [msg(S), 2]\ if \\ \big(\bigvee_{1 \leq i \leq n} \bigvee_{1 \leq j \leq k} \overline{x}_{m_i a_j} \geq 1)\big),$$

*where*

$$msg(S)(x_{m_i a_j}) = x_{m_i a_j} + \overline{x}_{m_i a_j}, \text{ for } 1 \leq i \leq n \text{ and } 1 \leq j \leq k$$

*and*

$$msg(S)(\overline{x}_{m_i a_j}) = 0, \text{ for } 1 \leq i \leq n \text{ and } 1 \leq j \leq k.$$

At this point, we simulate the effects of membrane dissolving. We have to drive the elements leaving the actual membranes by applications of $in_j$ or $out$ rules or elements of membranes freshly dissolved into membranes remaining existent after performing of the maximal parallel step. To this end, we define the following molecule.

**Definition 3.**

$$Dis_i = replace\ [S, 3]\ by\ [dis_i(S), 3]\ if \\ \big(x_{d_i} = 1 \wedge \\ \big(\bigvee_{1 \leq j \leq k} x_{m_i a_j} \geq 1)\big),$$

*where*

$$dis_i(S)(x_{m_j a_l}) = \begin{cases} x_{m_j a_l} + x_{m_i a_l} & \text{if } j = \mu(i), \\ 0 & \text{if } j = i, \\ x_{m_j a_l} & \text{otherwise.} \end{cases}$$

We also need some auxiliary molecules to set the values indicating the validity of rules to zero, in order to start a new maximal parallel step. Thus

**Definition 4.**

$$RemVal(r) = replace\ [S, 4]\ by\ [S[x_r/0], 4]\ if\ x_r = 1.$$

Now we are in a position to determine the molecule leading us through the simulation process. Let

$$\begin{aligned}
Val_\rho &= \bigcup\{Val_\rho(r) \mid r \in \mathcal{R}\},\\
App &= \bigcup\{App(r) \mid r \in \mathcal{R}\},\\
Dis &= \bigcup\{Dis_i \mid i \in \{1,\ldots,n\}\},\\
RemVal &= \bigcup\{RemVal(r) \mid r \in \mathcal{R}\},\\
Sync &= replace\ \langle [S, x_{sync}], Val_\rho, App, Msg, Dis, RemVal\rangle\ by\\
&\quad \langle [S, x_{sync} + 1\ mod\ 5], Val_\rho, App, Msg, Dis, RemVal\rangle\ if\\
&\quad \bigvee_{1\le i\le n} x_{r_i} = 1,\\
&\quad \text{where } S \text{ is a description pattern.}
\end{aligned}$$

**Notation 2** *Let $N$ be a molecule and let*

$$M(N) = (\langle N, Val_\rho, App, Msg, Dis\rangle, Sync).$$

*If $C$ is a configuration of $\Pi$ such that $C \Rightarrow^* C'$ for some $C'$ and $i \in \{0,1,2\}$, then we write*

$$M(C', i) = M([Descr_\mu(C'), i]).$$

The terms of the chemical calculus, and also the configurations of membrane systems can be considered as rewriting systems. A rewriting system, as used in this paper, is a pair $\mathcal{A} = \{\Sigma, (\rightarrow_i)_{i\in I}\}$, where $\Sigma$ is a set and $(\rightarrow_i)_{i\in I}$ is a set of binary relations defined on $\Sigma$. The relations $(\rightarrow_i)_{i\in I}$ are called reduction relations. It is supposed that a reduction relation $\rightarrow_i$ is compatible with the term formation rules. Moreover, if $\rightarrow_i$ is a reduction relation, we denote by $\rightarrow_i^*$ its reflexive, transitive closure. We may use the notation $\rightarrow = \cup_{i\in I}(\rightarrow_i)$, too. In the following, the set $\Sigma$ is the set of configurations of a P system or, in the case of the chemical formalism, the set of $\gamma$-terms, and $\rightarrow_i$ are the binary relations rendering configurations to configurations or terms to terms, respectively. We say that $m \in \Sigma$ is in *normal form*, if there is no $n \in \Sigma$, such that $m \rightarrow n$. Moreover, an $m \in \Sigma$ is *strongly normalizable*, if every reduction sequence starting from $m$ is finite, or *weakly normalizable*, if there exists a finite reduction sequence starting from $m$. We say that a molecule or a membrane $M$ is $\rightarrow_i$-irreducible, if there is no $M'$ such that $M \rightarrow_i M'$. In what follows, to conform to the usual membrane system notation, we use $\Rightarrow$ to denote $\rightarrow$ when we speak of a rewriting step in a membrane computation.

**Theorem 1.** *(1) Let $\Pi = (O, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n, \rho_1, \ldots, \rho_n)$ be a P system of order $n$ with membrane dissolving, promoter/inhibitor sets for rules and priority relations. Assume*

$$C_0 = (\mu, w_1, \ldots, w_n) \Rightarrow^* C_1 = (\mu', w'_{n_1}, \ldots, w'_{n_i}),$$

*where $1 \le n_1 \le \ldots \le n_i \le n$. Then*

$$M(C_0, 0) \rightarrow^* M(C_1, 0).$$

*If the computation starting from $C_0$ contains at least one step, then the reduction sequence starting from $M(C_0, 0)$ is non-empty either.*

*(2) Let $\mathit{\Pi}$ be a P system as above. Assume*

$$M(C_0, 0) \rightarrow^* M([N, 0]), \ \textit{and}$$

*assume that $\overline{c}_{ij} = 0$ for $(1 \leq i \leq n)$ and $(1 \leq j \leq k)$ in $N$ and $[N, 0]$ is $Val_\rho$ irreducible. Then there exists a configuration $C_1 = (\mu', w'_{n_1}, \ldots, w'_{n_i})$ of $\mathit{\Pi}$ such that $M([N, 0]) = M(C_1, 0)$ and*

$$C_0 \Rightarrow^* C_1.$$

*Moreover, if the length of $M_\mu(C_0) \rightarrow^* M([N, 0])$ is at least one, then the length of the computation starting from $C_0$ is non-zero.*

We work our way to the proof of the theorem by stating several auxiliary lemmas.

As formulated in [2], a computational step starting from a configuration $C_0$ of $\mathit{\Pi}$ consists of a maximal parallel step (mpr), a step for removing the directions from the targeted elements (tar) and a step for accomplishing membrane dissolution ($\delta$). In notation, if $C_0$ is a configuration of $\mathit{\Pi}$ and $C_0 \Rightarrow C_1$, then there are $C'_0$ and, if $\delta$ is present, $C''_0$ such that

$$C_0 \Rightarrow^*_{mpr} C'_0 \Rightarrow_{tar} C''_0 \Rightarrow_\delta C_1.$$

In the present paper, instead of $\Rightarrow_{tar}$, we choose a sequential relation (msg) defined in Definition 5 for removing messages instead of parallel communication rules, which equally suffices for our purposes. In what follows, if $C \Rightarrow_s C'$ by an intermediate step, we denote by $s \in mpr$ ($s \in msg, s \in \delta$) the fact whether $s$ is a maximal parallel, message removing, or membrane dissolving step, respectively.

We verify the lemmas simultaneously by induction on the number of intermediate steps in a computational step of the P system and on the number of reductions in the chemical calculus.

**Notation 3** *Let $C'$ be an (intermediate) configuration, where $C \Rightarrow^* C'$. Let $Descr_\mu(C')$ be the description of $C'$ relative to $\mu$. Then we use the notation below to extract the corresponding values from $M(C', l)$:*

$$\begin{array}{lll}
\lfloor M(C', l) \rfloor_{c_{ij}} & = Descr_\mu(C')_{i \cdot k + j} & (0 \leq i \leq n-1, \ 1 \leq j \leq k), \\
\lfloor M(C', l) \rfloor_{\overline{c}_{ij}} & = Descr_\mu(C')_{(n+i) \cdot k + j} & (1 \leq i \leq n, \ 1 \leq j \leq k), \\
\lfloor M(C', l) \rfloor_{d_i} & = Descr_\mu(C')_{2n \cdot k + i} & (1 \leq i \leq n), \\
\lfloor M(C', l) \rfloor_{r_{ij}} & = Descr_\mu(C')_{(2k+1) \cdot n + k_1 + \ldots + k_{i-1} + j} & (1 \leq j \leq k_i, \ 1 \leq i \leq n).
\end{array}$$

The following claims can be verified easily. Below, let $D$ denote a description.

*Claim.* Let $M([D, 0]) \rightarrow^*_{Val} M'$. Then $M' = M([D', 0])$, where $D'$ is a description.

*Claim.* Let $M([D, 1]) \rightarrow^*_{App} M'$. Then $M' = M([D', 1])$, where $D'$ is a description.

*Claim.* Let $M([D, 2]) \rightarrow^*_{Msg} M'$. Then $M' = M([D', 2])$, where $D'$ is a description.

*Claim.* Let $M([D, 3]) \rightarrow^*_{Dis} M'$. Then $M' = M([D', 3])$, where $D'$ is a description.

In the following, we assume that every possible configuration is the result of some computational sequence starting from a fixed configuration $C$ of the P system $\Pi = (O, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n, \rho_1, \ldots, \rho_n)$ of order $n$ with membrane dissolving, promoter/inhibitor sets for rules, and priority relations.

We prove the two parts of the theorem by simultaneous induction on the number of reduction steps in the chemical calculus and computational steps in the P system, respectively.

**Lemma 1.** *(1) If $C' \Rightarrow^*_{mpr} C''$, then $M(C',1) \rightarrow^*_{App} M(C'',1)$, and conversely,*

*(2) if we assume $M(C',1) \rightarrow^*_{App} M''$, then there is $C''$ such that $C' \Rightarrow^*_{mpr} C''$ and $M'' = M(C'',1)$.*

*Proof.* We prove the lemma by simultaneous induction on the lengths of the reduction sequences. Assume we know the result for reduction sequences of lengths at most $s$.

(1) Let $C \Rightarrow^* C'$, assume $C' = (\mu', w'_1, \ldots, w'_n)$. Suppose $C' \Rightarrow^s_{mpr} C''' \Rightarrow_r C''$, $C'' = (\mu'', w''_1, \ldots, w''_n)$, $C''' = (\mu''', w'''_1, \ldots, w'''_n)$ and $r = u \rightarrow v \in \mathcal{R}_i$. Since $r$ is applicable to $C'''$, we have $\lfloor M(C''',1) \rfloor_r = 1$ and $u(a_j) \leq w_i(a_j)$, which means $u(a_j) \leq \lfloor M(C''',1) \rfloor_{c_{ij}}$. These together imply that $App(r)$ can be applied to $M(C''',1)$ yielding $M([apply(Descr_\mu(C'''), r), 1])$.

- Let $\lfloor M([apply(Descr_\mu(C'''), r), 1]) \rfloor_{c_{lj}} = s_{lj}$. Then $s_{lj} = \lfloor M(C''',1) \rfloor_{c_{lj}} - u(a_j) = w'''_i(a_j) - u(a_j)$, if $l = i$, and $s_{lj} = \lfloor M(C''',1) \rfloor_{c_{lj}} = w'''_i(a_j)$ otherwise.
- Let $\lfloor M([apply(Descr_\mu(C'''), r), 1]) \rfloor_{\overline{c}_{lj}} = t_{lj}$. Then $t_{lj} = \lfloor M(C''',1) \rfloor_{\overline{c}_{lj}} + v(a_j, here)$, if $l = i$, $t_{lj} = \lfloor M(C''',1) \rfloor_{\overline{c}_{lj}} + v(a_j, in_h)$, if $l = h \neq i$ and $\mu(m_h) = m_i$, and $t_{lj} = \lfloor M(C''',1) \rfloor_{\overline{c}_{lj}} + v(a_j, out)$, if $l = \mu'''(i)$. Taking all these into account, $t_{ij} = w''_i(a_j, here) + \sum_{p \neq i, \mu(m_i) = m_p} w''_p(a_j, in_i) + \sum_{\mu(p) = i} w''_p(a_j, out)$ remains valid.
- If $v(\delta) = 1$, then $\lfloor M(C'', i) \rfloor_{d_i}$ is set to 1.

(2) Let $C \Rightarrow^* C'$, and $M(C',1) \rightarrow^*_{App} M''$. It is enough to prove the result for the case $M(C',1) \rightarrow_{App(r)} M''$, where $r = u \rightarrow v$. By Claim 3, $M'' = M([D'',1])$. Since $r$ is applicable to $M(C',1)$, we have, by $\lfloor M(C',1) \rfloor_r = 1$, that $r = u \rightarrow v \in \mathcal{R}_i$ is valid for some fixed $i$ depending on $r$. Moreover, $u(a_j) \leq \lfloor M(C',1) \rfloor_{c_{lj}} = w'_i(a_j)$, for every $1 \leq j \leq k$, which makes $r$ applicable to $C'$. From this point on, we can show by a reasoning similar to that of the previous point that $D'' = M(C'',1)$, where $C' \Rightarrow_r C''$. We omit the details. □

Instead of parallel communication as defined in [2] we choose the simpler way which is equally suitable to our present purposes and we define $\Rightarrow_{msg}$ as the following set of sequential multiset transformations.

**Definition 5.** *Let $C = (w_1, \ldots, w_n, \mu)$ and $C' = (w'_1, \ldots, w'_n, \mu, )$. Then $C \Rightarrow^*_{tar} C'$ holds iff one of the following cases is valid.*

1. *Assume that $w_i(a_j, here) > 0$. Then $w'_i(a_j) = w_i(a_j) + w_i(a_j, here)$ and $w'_i(a_j, here) = 0$. All the other values remain unchanged.*
2. *Assume $w_i(a_j, in_l) > 0$. Then $w'_l(a_j) = w_l(a_j) + w_i(a_j, in_l)$ and $w'_i(a_j, in_l) = 0$. All the other values remain unchanged.*
3. *Assume $w_i(a_j, out) > 0$ and $l = \mu(i)$ is defined. Then $w'_l(a_j) = w_l(a_j) + w_i(a_j, out)$ and $w'_i(a_j, out) = 0$. If $i = Skin$, then $w'_i(a_j, out) = 0$. All the other values remain unchanged.*

**Lemma 2.** *(1) Let $C' \Rightarrow^*_{msg} C''$, and assume that $C''$ is msg-irreducible. Then $M(C',2) \rightarrow_{Msg} M(C'',2)$.*

*(2) Conversely, assume $M(C',2) \rightarrow_{Msg} M''$. Then there is $C''$ such that $C' \Rightarrow_{msg} C''$, $C''$ is msg-irreducible, and $M'' = M(C'',1)$.*

*Proof.* We prove by induction on the number of steps in $C \Rightarrow_{msg} C'$ that, for every $1 \leq i \leq n$ and $1 \leq j \leq k$,

$$Descr_\mu(C)_{c_{ij}} + Descr_\mu(C)_{\overline{c}_{ij}} = Descr_\mu(C')_{c_{ij}} + Descr_\mu(C')_{\overline{c}_{ij}}. \tag{3}$$

To this end, we show that, if $C \rightarrow_{msg} C'$ and $C = (\mu, w_1, \ldots, w_n)$ and $C = (\mu, w'_1, \ldots, w'_n)$, then

$$\begin{aligned} &w_i(a_j) + w_i(a_j, here) + \sum_{p \neq i} w_p(a_j, in_i) + \sum_{\mu(p)=i} w_p(a_j, out) = \\ &w'_i(a_j) + w'_i(a_j, here) + \sum_{p \neq i} w'_p(a_j, in_i) + \sum_{\mu(p)=i} w'_p(a_j, out). \end{aligned} \tag{4}$$

We treat Point 2 of Definition 5, the remaining cases can be handled similarly. Let $C \Rightarrow_{msg} C'$ by Point 2 of Definition 5. Assume $w_i(a_j, in_l) > 0$. Let us consider only the case $i = l$ in Equation 4, since for all the other cases the equation trivially holds. But in this case the left hand side contains $w_l(a_j) + w_i(a_j, in_l)$, and the right hand side contains the corresponding $w'_l(a_j) + w'_i(a_j, in_l)$, which, by definition, are equal.

($\Rightarrow$) Let $C \Rightarrow_{msg} C'$, assume that $C'$ is msg-irreducible. A msg-irreducible P system with the *Skin* membrane as the outermost membrane contains no messages, thus, by Equation 3, $M(C,2) \rightarrow_{Msg} M(C',2)$.

($\Leftarrow$) Let $M(C,2) \rightarrow_{Msg} N'$. Then, by Claim 3, $N' = M(D',2)$ for some description $D'$. Let $C \Rightarrow_{msg} C'$ such that $C'$ is msg-irreducible. Then $C'$ is message free, which, by Equation 4, entails $D' = Descr_\mu(C')$.

□

Now, following [1], we define the skeleton of a configuration $(\mu, w_1, \ldots, w_n)$ as $U' = (u'_1, \ldots, u'_n)$, where $u'_i = *$, if membrane $i$ is dissolved or under dissolution (that is, $u_i(\delta) = 1$ and $i \neq Skin$) and $u'_i = 0$ otherwise. Let

$$\begin{aligned} &\mu^0(i) = i, \\ &\mu^j(i) = \mu(\mu^{j-1}(i)) \quad \text{for } j > 0. \end{aligned}$$

Let $\mu_{U'}(i) = min\{j \mid \mu^k(i) = j \wedge u'_j \neq * \wedge u'(\mu^l(i)) = * \text{ for } 0 \leq l \leq k-1\}$. That is, $\mu_{U'}(i)$ is the smallest membrane containing membrane $i$ which exists or does not disappear. Let $C' \Rightarrow_\delta C''$, assume $w'_l(\delta) = 1$ for at least one membrane $m_l$. We define the effect of the dissolution rule as follows: $(\mu', w'_1, \ldots, w'_n) \Rightarrow_\delta (\mu'', w''_1, \ldots, w''_n)$, where $w''_i = *$ provided $u'_i = *$, and $w''_i(a_j) = w'_i(a_j) + \sum\{w'_l(a_j) \mid \mu_{U'}(l) = i, w'_l(\delta) = 1\}$, if $u'(i) = 0$.

**Lemma 3.** *(1) If $C' \Rightarrow_\delta C''$, then $M(C',3) \rightarrow^*_{Dis} M(C'',3)$.*

*(2) Conversely, assume that $M(C',3) \rightarrow^*_{Dis} M''$, and $M''$ is Dis-irreducible. Then there exists a proper configuration $C''$ with $C' \Rightarrow_\delta C''$ and $M'' = M(C'',3)$.*

*Proof.* (1) Let $C' = (\mu, w'_1, \ldots, w'_n) \Rightarrow_\delta C''$, and assume that $w'_i(\delta) = 1$. Then $\lfloor M(C',3) \rfloor_{d_i} = 1$. Let $\lfloor M(C',3) \rfloor_{c_{ij}} > 1$ for some $1 \leq j \leq k$. Then $M(C',3) \rightarrow_{Dis_i} M([dis_i(Descr_\mu(C')),3])$. Let $p = \mu_{U'}(i)$, where $U'$ is the skeleton of $C'$. Let $D' = Descr_\mu(C')$ and $D'' = dis_i(Descr_\mu(C'))$. Let us denote by $D'_{c_{ij}}$ and $D''_{c_{ij}}$ the values of the descriptions pertaining to the coordinates $(i,j)$. It follows immediately, by Definition 3, that

$$\begin{aligned} & D'_{c_{pj}} + \sum\{D'_{c_{lj}} \mid \mu_{U'}(l) = p, D'_{d_l} = 1\} = \\ & D''_{c_{pj}} + \sum\{D''_{c_{lj}} \mid \mu_{U'}(l) = p, D''_{d_l} = 1\}. \end{aligned} \tag{5}$$

In other words, for every $1 \leq j \leq k$, the sums of the occurrences of elements $a_j$ in the dissolved or to be dissolved descendants of membrane $m_p$ plus the multiplicity of $a_j$ in $m_p$ remain the same at a dissolution step in the chemical calculus. Let $M(C',3) \rightarrow_{Dis} M([D,3])$ such that $M([D,3])$ is irreducible with respect to $Dis$. Then $D_{d_i} = 1$ implies $D_{c_{ij}} = 0$, which, by Equation 5, involves that $D = Descr_\mu(C'')$.

(2) Let $M(C',3) \rightarrow^*_{Dis} M''$. By Claim 3, there exists a description $D''$ such that $M'' = M([D'',3])$. Since $M''$ is Dis irreducible, $D''_{d_i} = 1$ implies $D''_{c_{ij}} = 0$. This means, there is a proper configuration $C''$ such that $Descr_\mu(C'') = D''$. Assume $w'_s(\delta) = 1$ holds in $C'$. Let $D' = Descr_\mu(C')$. Let $U'$ be the skeleton of $D'$ and $p = \mu_{U'}(s)$. Since $M''$ is Dis irreducible, Equation 5 simplifies to

$$D'_{c_{pj}} + \sum\{D'_{c_{lj}} \mid \mu_{U'}(l) = p, D'_{d_l} = 1\} = D''_{c_{pj}}.$$

Taking the corresponding configurations, this amounts to $C' \Rightarrow_\delta C''$. □

*Proof of Theorem 1.*

(⇒)Let $\Pi = (O, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n, \rho_1, \ldots, \rho_n)$ be a P system of order $n$ with membrane dissolving, promoter/inhibitor sets for rules and priority relations. Assume $C_0 \Rightarrow^t C_1$. We prove by induction on $t$ that $M(C_0,0) \rightarrow^* M(C_1,0)$. Let $C \Rightarrow^{t-1} C_2 \Rightarrow C_1$. Assume $C_2 = (\mu'', w''_{n_1}, \ldots, w''_{n_i})$, $C_1 = (\mu', w'_{n_1}, \ldots, w'_{n_i})$. Assume there exists $N''$ such that $M(C'',0) \rightarrow_{Val} N''$. But $Val_\rho(r)$ is applicable iff $r$ is valid and no rule $r'$ with $(r', r) \in \rho$ is valid, this means $N'' = M(C'',0)$ and $M(C'',0)$ is $Val_\rho$ irreducible. In this case

$$M(C'',0) \rightarrow_{Sync} M(C'',1).$$

Putting Lemmas 1, 2 and 3 together, taking into account the fact that $M(E,i) \rightarrow_{Sync} M(E, i+1 \ mod\ 5)$ whenever $M(E,i)$ is irreducible for the corresponding reduction, we obtain that there exists a configuration $\widetilde{C}$ and a description $D$ such that

$$M(C'',0) \rightarrow^*_{RemVal} M([D,4]) \rightarrow_{Sync} M([D,0]),$$

where $D$ is the description $Descr_\mu(\widetilde{C})$ except for the values $D_r = 0$. A transition in $Val_\rho(r)$ is applicable at most twice for every rule $r$. This means there is a description $D'$ such that

$$M([D,0]) \rightarrow^*_{Val_\rho} M([D',0])$$

and $M([D',0])$ is $Val_\rho$ irreducible. But then $D' = Descr_\mu(\widetilde{C})$.

($\Leftarrow$)Follows in a way similar to the above part from Lemmas 1, 2 and 3, but this time applying the other directions of the lemmas.

**Corollary 1.** *Let $\Pi = (O, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n, \rho_1, \ldots, \rho_n)$ and let $C = (\mu, w_1, \ldots, w_n)$. Then $\Pi$ is strongly (resp. weakly) normalizing iff $M(C, 0)$ is strongly (resp. weakly) normalizing. Moreover, the halting computations starting from $C$ provide the same results as those supplied by the terminating reduction sequences of $M(C, 0)$.*

## References

1. O. Agrigoroaiei, G. Ciobanu, Flattening the transition P systems with dissolution. In: M. Gheorghe, T. Hinze, Gh. Păun, G. Rozenberg, A. Salomaa (editors), *Membrane Computing, 11th International Conference, CMC 2010, Jena, Germany, 2010, Revised Selected Papers.* Volume 6501 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg (2011) 53–64.
2. O. Andrei, G. Ciobanu and D. Lucanu, Structural operational semantics of P systems. In: R. Freund, Gh. Păun, G. Rozenberg, A. Salomaa (editors), *Membrane Computing, 6th International Workshop, WMC 2005, Vienna, Austria, 2005, Revised Selected and Invited Papers.* Volume 3850 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg (2006) 32–49.
3. O. Andrei, G. Ciobanu and D. Lucanu, A rewriting logic framework for operational semantics of membrane systems. *Theoretical Computer Science*, 373 (2007) 163–181.
4. J.P. Banâtre, P. Fradet, Y. Radenac, Principles of chemical computing. *Electronic Notes in Theoretical Computer Science* 124 (2005) 133–147.
5. J.P. Banâtre, P. Fradet, Y. Radenac, Generalized multisets for chemical programming. *Mathematical Structures in Computer Science* 16 (2006) 557–580.
6. J.P. Banâtre, D. Le Métayer, A new computational model and its discipline of programming. *Technical Report RR0566*, INRIA (1986).
7. P. Battyányi, Gy. Vaszil, Describing membrane computations with a chemical calculus. *Fundamenta Informaticae*, 134 (2014) 39–50.
8. P. Bottoni, C. Martín-Vide, G. Păun, G. Rozenberg, Membrane systems with promoters/inhibitors. *Acta Informatica*, 38 (2002) 695–720.
9. G. Păun, Computing with membranes. *Journal of Computer and System Sciences*, 61(1) (2000) 108–143.
10. G. Păun, *Membrane Computing. An Introduction.* Springer, 2002.
11. G. Păun, G. Rozenberg, A. Salomaa (eds), *The Oxford Handbook of Membrane Computing.* Oxford University Press, 2010.