Metabolic Algorithm with Time-varying Reaction Maps

Luca Bianco, Federico Fontana, Vincenzo Manca

University of Verona
Department of Computer Science
strada Le Grazie, 15
37134 Verona, Italy
E-mail: {bianco,fontana}@sci.univr.it, vincenzo.manca@univr.it

Summary. A symbolic-based approach to modelling biochemical processes and cellular dynamics is likely to turn useful in computational biology, where attempts to represent the cell as a huge, complex dynamic system must trade with the linguistic nature of the DNA and the individual behavior of the organelles living within. The early version of the metabolic algorithm gave a first answer to the problem of representing oscillatory biological phenomena, so far being treated with traditional (differential) mathematical tools, in terms of rewriting systems. We are now working on a further version of this algorithm, in which the rule application is tuned by reaction maps depending on the specific phenomenon under consideration. Successful simulations of the Brusselator, the Lotka-Volterra population dynamics and the PKC activation foster potential applications of the algorithm in systems biology.

1 Introduction

Symbolic rewriting has traditionally been used to study and classify formal languages [18]. It was some years after Chomsky's fundamental discoveries that rewriting systems began to be applied to the study of the growth of some simple organisms and to the analysis of biological structures [9, 15].

These early applications of rewriting to practical case studies taken from the real world demonstrated the potential ability of a properly defined formal construct to represent, in principle, the *development* of at least some biological species. Such constructs, in fact, move step by step toward the definition of a language/structure until their computation terminates, hence their application to species in development emphasized the possibility for formal systems to figure out not only classes of languages, but also the paths along which their final structure takes form during the system evolution.

Recently, a research line has started which focuses on the rewriting system dynamic activity instead of its expressive power evaluated in terms of language types [20, 3, 11]. This line has been stimulated in an attempt to capture, by means of rewriting systems, the dynamics of a biochemical process. In this attempt a novel construct known as *P system* has come useful, provided its capability to represent several structural aspects of the cell along with many intra- and extra-cellular communication mechanisms [16, 4]. Such a dynamic perspective on rewriting employing P systems has already led to alternative representations of different biological dynamics [1, 19] and to new models of important pathological processes [3, 14].

In [3, 11] we have started to develop a *metabolic algorithm* that introduced a new perspective in the rewriting mechanism of P systems:

1. rules are not applied to objects. Rather, they are applied to *populations* of objects;
2. rules are specified along with *reactivities*. Every reactivity denotes the ability of the corresponding rule to compete against other rules in capturing part of a population, on which the reaction is performed.

We have gone further in this perspective, by associating every reactivity to a map that depends on the state of the system. Moreover we have added a strategy for partitioning the objects in the system at every transition, depending on the relative magnitude of every reactivity.

The performance shown by the metabolic algorithm in the simulation of well-known biochemical models, such as the Lotka-Volterra population dynamics [10, 21], the BZ chemical reaction [5], and the PKC activation process [2], fosters potential applications in critical open problems dealt with by systems biology [7]. Simulation in progress are confirming the effectiveness of the algorithm in modelling even more complex biochemical processes, such as those that evoke circadian rhythms in living bodies [8].

2 Metabolic Algorithm

As we have told in the introduction, the *metabolic algorithm* is built on P systems. For the sake of simplicity here, and in the following, we hypothesize that this system is made of just one membrane. In Section 5 we will briefly discuss the formal extensions needed to cope with more membranes.

To provide our algorithm with flexibility we will guarantee the fulfillment of the two following principles at any transition of a P system Π, working on the alphabet $\mathcal{A} = \{X, Y, \ldots, Z\}$ and provided with rules $r, s, \ldots, w \in R$:

- increasing the activity of a rule implies a proportional decrease of the rewriting activity of other rules sharing the same symbols. This condition reflects the concurrency among rules over a finite set of objects in the system;
- the applicability of rules is limited by those objects, whose availability in the system is low. This condition reflects a constraint on resources.

In the following we will reformulate these two points in quantitative terms.

In the early version of the metabolic algorithm we had postulated that proper reactivity constants affected the rewriting activity of every rule, respectively, in a way that a larger reactivity constant defined a higher rewriting activity of the corresponding rule. Then, this activity was properly *limited* by defining a population on which a rule $r : \alpha_r \rightarrow \beta_r$, transforming the string $\alpha_r \in \mathcal{A}^*$ into a new string $\beta_r \in \mathcal{A}^*$, could be applied during a transition depending on the number of objects available in the system immediately before that transition.

The new version of the metabolic algorithm requires, firstly, to recognize the *state* of the system. This state is used inside so-called *reaction maps* which generalize the former reactivity constants into time-varying functions. Once we have such maps at hand we will let the rules work according to the relative reactivity expressed by every map, meanwhile *limiting* this power to avoid over-consumption of the objects in the system. Finally, a simple stochastic method will be adopted to decide how to treat individual objects in the system, for which the procedure described so far does not take a definite decision. More in general, introducing stochastic properties in an algorithm can turn out to be particularly desirable when the dynamics is highly influenced by few molecules [6].

2.1 State of the System

In classical dynamic systems the values assumed along time by every variable usually form the state of the system. In a similar way, here we postulate that at every discrete time t the number of objects of each type is well defined for every membrane. Formally, the state of the system at time t is identified by a function

$$q_t : \quad \mathcal{A} \longrightarrow \mathbb{N}, \tag{1}$$

where $\mathcal{A}$ is the alphabet of the P system.

For instance, $q_t(X)$ gives the amount of objects X available in the system at time t. Note that we will usually omit to denote t, except for those cases in which specifying the time turns out to be convenient.

The state, hence, can be read by applying q_t to every symbol of the alphabet. The set of all states assumed along time by the system is expressed by:

$$Q = \{q_t \,|\, t \in \mathbb{N}\}. \tag{2}$$

This set, then, contains the complete information on the evolution of the system. Further insight on the state is not possible since q_t is the only probe we can use to observe it.

2.2 Reaction Maps

As opposite to the early metabolic algorithm [3] in which the reactivity constants had a direct (and time invariant) role on rewriting, here we generalize such constants into *reaction maps*, one for each rule, in a way that every reaction map gives the reactivity that the corresponding rule has when the system is in a given state. It follows that such maps are time varying, i.e., they in general specify different reactivities in correspondence of different temporal steps.

Formally speaking, for each rule r we define a reaction map F_r that maps states into real numbers:

$$F_r : \quad Q \longrightarrow \mathbb{R}. \tag{3}$$

Since q is defined at any temporal step, the application of a reaction map F_r ultimately results in a positive real number that we will take as the *reactivity of* r *in* q.

Such maps allow for a wide choice of possible definitions depending on the biological phenomenon under analysis: according to the traditional formulation of dynamic system it is not restrictive to consider real functions that in their structure include the state of the system plus factors such as the reactivity constants mentioned at the beginning of this section.

As an example, consider a rewriting system having an alphabet made of five symbols, $\mathcal{A} = \{A, B, C, D, E\}$, and two rules, r and s:

$$\begin{array}{rrcl} r: & ABB & \overset{k_r}{\rightarrow} & AC \\ s: & AE & \overset{k_s}{\rightarrow} & BD \end{array} \tag{4}$$

in which, consistently with notations traditionally adopted in biochemistry, we have specified constant reactivities (k_r and k_s) that are peculiar to each rule—they could be, for example, kinetic parameters related to the chemical reactions respectively associated to these rules.

Possible structures of the reaction maps might be the following ones:

- simple reactivity constants

$$\begin{aligned} F_r &= k_r \\ F_s &= k_s \end{aligned}$$

- reactivities driven by the law of mass action

$$\begin{aligned} F_r &= k_r \, q(A)q(B) \\ F_s &= k_s \, q(A)q(E) \end{aligned}$$

- reactivities depending on only the largest number of objects in the system that are visible to the rule

$$\begin{aligned} F_r &= \max\{q(A), q(B)\} \\ F_s &= \max\{q(A), q(E)\} \end{aligned}$$

- reactivities depending on an external promoter, like an enzyme capable of activating the reaction

$$\begin{aligned} F_r &= q(D) \\ F_s &= \{q(D)\}^2 \end{aligned}$$

In the following we will pick up this example as long as we need to illustrate the principles of the algorithm.

2.3 Reaction Weights

Reaction maps are not used directly, as reactivities. Rather, their activity is proportionally distributed among the rules by means of so-called *reaction weights.* Every reaction weight then gives, for each symbol, a population amount a rule applies to in order to proportionally consume the corresponding object. By denoting with $\alpha(i)$ the ith symbol in a string α, with $|\alpha|$ the length of the same string, and with $|\alpha|_X$ the number of occurrences of X in α, then we define the reaction weight $W_r\big(\alpha_r(i)\big)$ for $r:\ \alpha_r \to \beta_r$ with respect to the symbol $\alpha_r(i)$.

Normalization can be straightforwardly expressed in quantitative terms if we think that all rules co-operate, each one with its own reactivity, to consume all available objects. Thus, it must be:

$$\sum_{\rho \in R \,|\, X \in \alpha_\rho} W_\rho\big(X\big) = 1 \quad \forall\, X \in \mathcal{A} \tag{5}$$

that is, for each symbol the sum of the reaction weights made over the rules containing that symbol in their left part equals unity.

Holding this constraint, we can define the reaction weights for each $r \in R$ as

$$W_r\big(\alpha_r(i)\big) = \frac{F_r}{\displaystyle\sum_{\rho \in R \,|\, \alpha_r(i) \in \alpha_\rho} F_\rho} \quad , \quad i = 1, \dots, |\alpha_r| \tag{6}$$

Note that, similarly to what happens in (5), we sum at the denominator over the rules containing the symbol $\alpha_r(i)$ in their left part.

Returning to our example, we have to compute $W_r(A)$, $W_r(B)$, $W_s(A)$, $W_s(E)$:

$$\begin{aligned} W_r(A) &= \frac{F_r}{F_r + F_s}\ W_r(B) = \frac{F_r}{F_r} = 1 \\ W_s(A) &= \frac{F_s}{F_r + F_s}\ W_s(E) = \frac{F_s}{F_s} = 1 \end{aligned} \tag{7}$$

2.4 Limitation, Rounding and State Transition

For what we have said in the above, the available objects are consumed proportionally to the reaction weights. Then, in our example we have to choose whether to consume

$$W_r(A)q(A) \quad \text{or} \quad W_r(B)q(B) \tag{8}$$

objects using r, and

$$W_s(A)q(A) \quad \text{or} \quad W_s(E)q(E) \tag{9}$$

using s—provided for simplicity that all values in (8) and (9) are integer.

The right choice is figured out by considering that every rule cannot consume more than the amount of the (reactant) object, *taken with its own multiplicity in the reaction*, whose availability in the system is lowest. Limitation, then, comes out for every rule by minimizing among all reactants participating to it:

$$\Lambda_r = \min_{i=1,\ldots,|\alpha_r|} \left\{ W_r\big(\alpha_r(i)\big) \frac{q\big(\alpha_r(i)\big)}{\big|\alpha_r\big|_{\alpha_r(i)}} \right\} \tag{10}$$

Still, Λ_r is a real number. As opposite to this, a genuine object-based rewriting system must restrict the rule application domain to integer values. Instead of, for instance, rounding the minima obtained by (10), we prefer the following policy (later we will understand why): for every rule, compare the fractional part $\mathrm{frac}(\Lambda_r)$ of Λ_r to a random variable v_r defined between 0 and 1, and choose the floor of Λ_r if this fraction is smaller, the ceiling otherwise. In this way, new rounded minima result to be equal to:

$$\overline{\Lambda}_r = \begin{cases} \mathrm{floor}(\Lambda_r) \ , \mathrm{frac}(\Lambda_r) \le v_r \\ \mathrm{ceil}(\Lambda_r) \quad , \mathrm{frac}(\Lambda_r) > v_r \end{cases} . \tag{11}$$

As a result of this step we obtain the set $\{\overline{\Lambda}_r, r \in R\}$, containing the number of objects each rule will be applied to.

In conclusion, for every symbol $X \in \mathcal{A}$ the change in the number of objects due to r is equal to the *stoichiometric factor* of r, equal to $|\beta_r|_X - |\alpha_r|_X$, times the value $\overline{\Lambda}_r$:

$$\Delta_r(X) = \overline{\Lambda}_r \left(|\beta_r|_X - |\alpha_r|_X\right) \tag{12}$$

It descends that for every symbol $X \in \mathcal{A}$ the state evolves according to the following formula:

$$q_{t+1}(X) = q_t(X) + \sum_{r \in R} \Delta_r(X) \tag{13}$$

Again in our example, let us suppose that at time t it is $q(A) = q(B) = q(C) = q(D) = q(E) = \tilde{q}$, furthermore $F_r = 3/4\, F_s$. Then,

$$\Lambda_r = \min\left\{ \frac{3/4F_s}{3/4F_s + F_s}\, \tilde{q}, \frac{1}{2}\, \tilde{q} \right\} = \frac{3}{7}\, \tilde{q}$$

$$\Lambda_s = \min\left\{ \frac{F_s}{3/4F_s + F_s}\, \tilde{q}, \tilde{q} \right\} = \frac{4}{7}\, \tilde{q}$$

After rounding Λ_r and Λ_s (here, for simplicity, we suppose to have found integers already at the limitation step) we have

$$\Delta_r(A) = \frac{3}{7}\, \tilde{q}\, (1 - 1) = 0$$

$$\Delta_r(B) = \frac{3}{7}\, \tilde{q}\, (0 - 2) = -\frac{6}{7}\, \tilde{q}$$

$$
\begin{aligned}
\mathit{\Delta}_r(C) &= \frac{3}{7}\,\tilde{q}\,(1-0) = \frac{3}{7}\,\tilde{q}\\
\mathit{\Delta}_s(A) &= \frac{4}{7}\,\tilde{q}\,(0-1) = -\frac{4}{7}\,\tilde{q}\\
\mathit{\Delta}_s(B) &= \frac{4}{7}\,\tilde{q}\,(1-0) = \frac{4}{7}\,\tilde{q}\\
\mathit{\Delta}_s(D) &= \frac{4}{7}\,\tilde{q}\,(1-0) = \frac{4}{7}\,\tilde{q}\\
\mathit{\Delta}_s(E) &= \frac{4}{7}\,\tilde{q}\,(0-1) = -\frac{4}{7}\,\tilde{q}
\end{aligned}
$$

in a way that

$$
\begin{aligned}
q_{t+1}(A) &= \tilde{q} - \frac{4}{7}\,\tilde{q} = \frac{3}{7}\,\tilde{q}\\
q_{t+1}(B) &= \tilde{q} + \Big(-\frac{6}{7} + \frac{4}{7}\Big)\,\tilde{q} = \frac{5}{7}\,\tilde{q}\\
q_{t+1}(C) &= \tilde{q} + \frac{3}{7}\,\tilde{q} = \frac{10}{7}\,\tilde{q}\\
q_{t+1}(D) &= \tilde{q} + \frac{4}{7}\,\tilde{q} = \frac{11}{7}\,\tilde{q}\\
q_{t+1}(E) &= \tilde{q} - \frac{4}{7}\,\tilde{q} = \frac{3}{7}\,\tilde{q}
\end{aligned}
$$

From the last equations it follows that

$$\sum_{X\in\mathcal{A}} q_{t+1}(X) = 32/7\,\tilde{q} < \sum_{X\in\mathcal{A}} q_t(X) = 5\tilde{q}$$

Interesting to see, in this system the total number of objects cannot increase along time. In other words in our example the following relation holds:

$$\sum_{X\in\mathcal{A}} q_{t+1}(X) \le \sum_{X\in\mathcal{A}} q_t(X)$$

3 Flexibility of the Algorithm

The proposed algorithm has two basic access points where parameters can be put into: the reaction maps, and the stochastic properties of v_r.

- Reaction maps can be defined with relative freedom, and even changed during the process according to the specific phenomenon under study. Their activity, in fact, is in any case normalized by the reaction weights. Occasionally some maps may result in null values: in this case reaction weights might arise in the form 0/0, and proper strategies must be put into action to handle them properly. Reaction maps, in conclusion, enable the fine control of the macroscopic, i.e., *deterministic* part of the process.

- Conversely, the statistics of v_r has consequences on the system behavior that become as more important, as fewer objects are present in the system. In other words it influences the *stochastic* part of the process, i.e., its unpredictability in front of individual drifts from the average behavior. Although further research must be carried out to shift the metabolic algorithm closer to stochastic methods used in biochemistry [17], nevertheless the control of v_r already allows to handle, at least to some extent, an interesting property of most discrete population dynamics, according to which the decision taken by an individual becomes as more crucial, as less populated the system is [6]. This feature is evident in the simulation of the Lotka-Volterra dynamics proposed in the following.

  The rounding policy expressed by (11) does not prevent that the resulting application of rules exceeds the available resources in the system. As an example suppose that, during a transition, it happens that $\overline{\Lambda}_r \geq \Lambda_r$ for each $r \in R$: in this case it is likely that the consequent application of the rules over-consumes at least some objects available in the system. To prevent this we must check that

  $$\sum_{r \in R} \overline{\Lambda}_r |\alpha_r|_X \leq q(X) \quad \forall X \in \mathcal{A} \tag{14}$$

  otherwise the set of minima must be computed again.

  In first approximation v_r can be chosen to have a uniform distribution. We will present here an example in which a different choice of the random variable leads to more accurate simulation results.

3.1 Transparent Rules

The metabolic algorithm allows to tune the activity of rules. Tuning is achieved by adding in the system so-called *transparent rules*, i.e., rules in the form $\alpha_r \to \alpha_r$. Proper reaction maps can be selected to put such transparent rules in concurrence with the other, effective rules sharing common reactants. In this way, during a transition of the system every rule is applied as less intensively, as larger the reactivity value expressed by a concurrent transparent rule is.

For instance, let us add a rule $t : A \xrightarrow{k_t} A$ in the system expressed by (4). This leads to the following reaction weights:

$$\begin{aligned} W_r(A) &= \frac{F_r}{F_r + F_s + F_t} \; W_r(B) = \frac{F_r}{F_r} = 1 \\ W_s(A) &= \frac{F_s}{F_r + F_s + F_t} \; W_s(E) = \frac{F_s}{F_s} = 1 \end{aligned} \tag{15}$$

Note that we can omit to compute $W_t(A)$ due to the transparency of the corresponding rule.

Clearly, F_t tunes the action of r and s over the symbol A. In the limit case $F_t = \infty$ the rule t inhibits the action of r and s over A, since in this case we have $W_r(A) = W_s(A) = 0$. Finally, a rule $t : ABE \to ABE$ would inhibit both r and s in the same limit case.

Transparent rules add further flexibility to the algorithm. In particular, they allow to observe the system evolution with the desired degree of resolution *regardless of any consideration about the granularity of the temporal step*. Changes in resolution are instead obtained by "hiding" objects to the system evolution by means of transparent rules. The way transparent rules work reflects an inherent attitude of the metabolic algorithm to scale its own resolution not along the time dimension, i.e., by means of a temporal scaling factor as it happens in most numerical methods. Rather, resolution is scaled by adapting the size of populations to the degree of precision expected for the experiment.

4 Results

We show results coming from the predator-prey population dynamics, the Brusselator, and the PKC activation process.

4.1 Predator-Prey Population Dynamics

The classic Lotka-Volterra population dynamics [10, 21] can be described by a simple set of rewriting rules in which X are preys and Y predators:

$$\begin{array}{ll} r: \;\; X \xrightarrow{k_r} XX & \text{prey reproduction} \\ s: XY \xrightarrow{k_s} YY & \text{predator reproduction} \\ t: \;\; Y \xrightarrow{k_t} \;\; \lambda & \text{predator death} \end{array} \qquad (16)$$

Here, we can tune the activity of every rule by selecting proper reactivity constants k_r, k_s and k_t proportional to the rate of reproduction and death of both predators and preys. We postulate F_s to be constantly proportional to k_s times the maximum number between preys and predators, $\max\{q(X), q(Y)\}$, that are present in the population at any system transition. Conversely, the remaining reaction maps are set to be constantly equal to the corresponding reactivity constants:

$$\begin{aligned} F_r &= k_r \\ F_s &= k_s \max\{q(X), q(Y)\} \\ F_t &= k_t \end{aligned} \qquad (17)$$

Moreover we add transparent rules accounting for preys that are *not* reproducing or being consumed and for predators that are *not* eating or dyeing:

$$\begin{array}{ll} u: X \xrightarrow{k_u} X & \text{prey standing by} \\ v: Y \xrightarrow{k_v} Y & \text{predator standing by} \end{array} \qquad (18)$$

The set of reaction weights is then equal to

$$
\begin{aligned}
W_r(X) &= \frac{F_r}{F_r + F_s + F_u} \\
W_s(X) &= \frac{F_s}{F_r + F_s + F_u} \quad W_s(Y) = \frac{F_s}{F_s + F_t + F_v} \\
& \qquad\qquad\qquad\quad W_t(Y) = \frac{F_t}{F_s + F_t + F_v}
\end{aligned}
\tag{19}
$$

Noticing that $|\alpha_s|_X = |\alpha_s|_Y = 1$, then the minimum between weighted preys and weighted predators must be calculated at each system transition as

$$
\begin{aligned}
\Lambda_s = \min\Big\{ & \frac{k_s \max\{q(X), q(Y)\}}{k_r + k_u + k_s \max\{q(X), q(Y)\}}\, q(X), \\
& \qquad \frac{k_s \max\{q(X), q(Y)\}}{k_t + k_v + k_s \max\{q(X), q(Y)\}}\, q(Y) \Big\}
\end{aligned}
\tag{20}
$$

If we in particular choose $k_r = k_t$ and $k_u = k_v$ this minimum becomes equal to

$$
\begin{aligned}
\Lambda_s &= \frac{k_s \max\{q(X), q(Y)\}}{k_r + k_u + k_s \max\{q(X), q(Y)\}} \min\{q(X), q(Y)\} \\
&= \frac{k_s\, q(X) q(Y)}{k_r + k_u + k_s \max\{q(X), q(Y)\}}
\end{aligned}
\tag{21}
$$

In this case we have

$$
\begin{aligned}
\Lambda_r &= \frac{k_r}{k_r + k_u + k_s \max\{q(X), q(Y)\}}\, q(X) \\
\Lambda_s &= \frac{k_s}{k_r + k_u + k_s \max\{q(X), q(Y)\}}\, q(X) q(Y) \\
\Lambda_t &= \frac{k_r}{k_r + k_u + k_s \max\{q(X), q(Y)\}}\, q(Y)
\end{aligned}
\tag{22}
$$

Interesting to notice, these values resemble the terms proposed by Lotka-Volterra in its model of population dynamics. The choice made (not by chance) in (17), then, adds insight on the meaning of the population dynamics equations in the context of a rewriting system whose objects play the role of either preys or predators.

Finally we check that the rounding procedure, making use of uniform random variables, does not produce over-consuming applications of the rules:

$$
\begin{aligned}
\overline{\Lambda}_r + \overline{\Lambda}_s &\le q(X) \\
\overline{\Lambda}_s + \overline{\Lambda}_t &\le q(Y)
\end{aligned}
\tag{23}
$$

(though, the existence of u and v makes this possibility remote).

A plot of the initial dynamic behavior of the predator-prey model is depicted in Figure 1 and, after 136000 observation slots, in Figure 2. These plots come out when we set $k_r = k_t = 3 \cdot 10^{-2}$, $k_s = 4 \cdot 10^{-5}$, and $k_u = k_v = 5$. Using

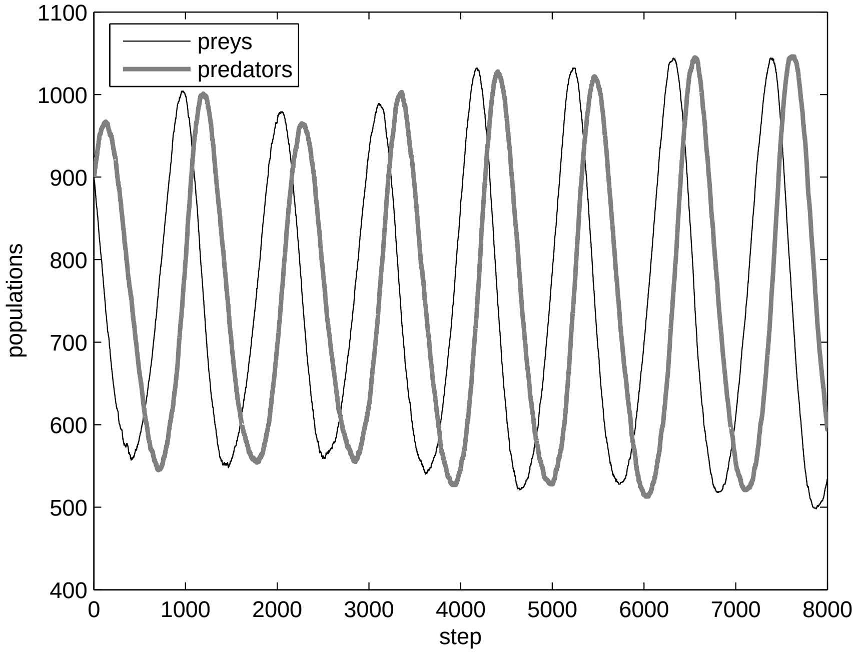

Fig. 1. Predator-prey initial dynamics.

these parameters, along with initial conditions $q(X) = q(Y) = 900$, the system exhibits an interesting oscillating behavior. The oscillation can evolve to the death of both species, as in Figure 2, or to the death of the predators solely. The long-term evolution in fact depends on single events taking place when few individuals, either preys or predators, are present in the system. Such a long-term behavior emphasizes the importance of a careful description of not only the reactivities, but also the relationships existing between individuals: the nature of these relationships can completely change the overall system evolution.

4.2 Brusselator

The Belousov-Zhabotinskii (BZ) reaction has represented a milestone in the history of physical chemistry, as it disclosed the previously unrecognized existence of oscillatory chemical phenomena [5]. A simple model of the BZ reaction is realized by the Brusselator [20]:

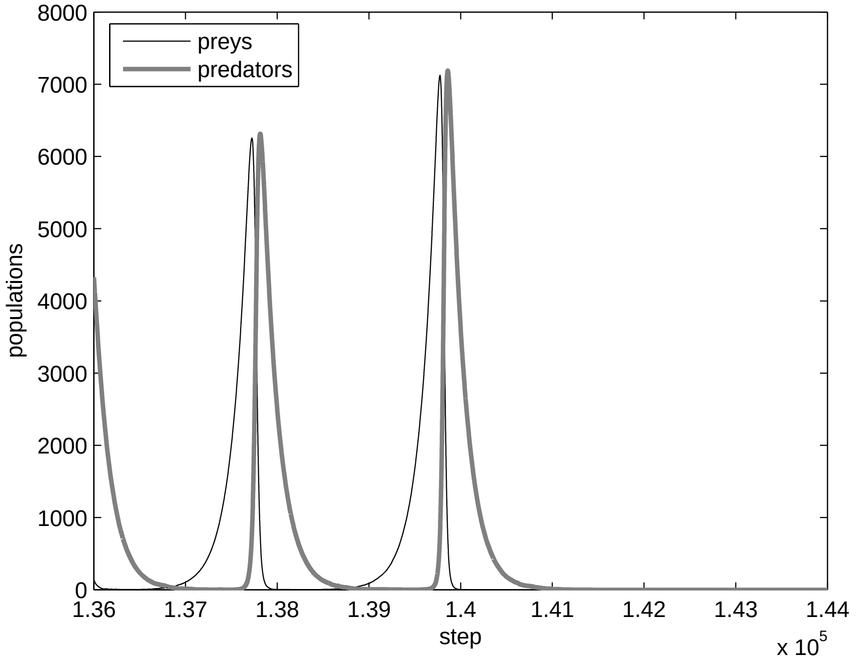

Fig. 2. Predator-prey dynamics after 136000 observation slots.

$$\begin{array}{rcl l}
r: & \lambda & \xrightarrow{k_r} & X & \text{reactant in}\\
s: & XXY & \xrightarrow{k_s} & XXX & \text{compound into reactant}\\
t: & X & \xrightarrow{k_t} & Y & \text{reactant into product}\\
u: & X & \xrightarrow{k_u} & \lambda & \text{reactant dissolving}
\end{array} \tag{24}$$

This set of rules accounts for the fact that the reactant X either turns into a product Y or participates in transforming the product back to the reactant itself. As opposite to the predator-prey model, the Brusselator includes a constant incoming and dissolving of reactant in the system.

The literature on the Brusselator suggests that the reaction activity depends on the concentrations of chemical elements according to the *law of mass action.* We then define the following reaction maps:

$$\begin{aligned}
F_r &= k_r\\
F_s &= k_s\,\{q(X)\}^2 q(Y)\\
F_t &= k_t\,q(X)\\
F_u &= k_u\,q(X)
\end{aligned} \tag{25}$$

It must be noticed that rules containing λ in their left part do not compete for a limited availability of reactant or for a bounded population, by definition. For this reason they act unconstrained, i.e., no reaction weight holds for any of them. Hence, we come up with the following reaction weights in which r is not considered:

$$\begin{aligned} W_s(X) &= \frac{F_s}{F_s + F_t + F_u} \;\; W_s(Y) = \frac{F_s}{F_s} = 1 \\ W_t(X) &= \frac{F_t}{F_s + F_t + F_u} \\ W_u(X) &= \frac{F_u}{F_s + F_t + F_u} \end{aligned} \tag{26}$$

Again we have to minimize only over s at any system transition:

$$\Lambda_s = \min\left\{ \frac{1}{|\alpha_s|_X} \frac{F_s(X)}{F_s(X) + F_t(X) + F_u(X)}\, q(X),\, q(Y) \right\} \tag{27}$$

so that in the end we have

$$\begin{aligned} \Lambda_r &= k_r \\ \Lambda_s &= q(Y) \min\left\{ \frac{1}{2} \frac{k_s}{k_t + k_u + k_s\, q(X) q(Y)} \{q(X)\}^2,\, 1 \right\} \\ \Lambda_t &= \frac{k_t}{k_t + k_u + k_s\, q(X) q(Y)}\, q(X) \\ \Lambda_u &= \frac{k_u}{k_t + k_u + k_s\, q(X) q(Y)}\, q(X) \end{aligned} \tag{28}$$

Finally we check out that the rounding procedure does not produce overconsuming applications of rules. By (14) it turns out that checking over X is sufficient:

$$2\overline{\Lambda}_s + \overline{\Lambda}_t + \overline{\Lambda}_u \leq q(X) \tag{29}$$

A plot of the dynamic behavior of our rewriting system modelling the Brusselator, in which we have set $k_r = 10$, $k_s = 9$, $k_t = 200$, $k_u = 5$, and initially $q(X) = 1$ and $q(Y) = 10$, is depicted in Figure 3. The overall behavior is satisfactory if compared to real experiments conducted over the BZ reaction [5].

An artifact, visible in the center part of the plot, arises after around 1350 steps consisting in a constant climb of the concentration of the reactant. This fact reveals the existence of periods in which the reaction goes in a stand-by situation, and then restarts with the oscillatory dynamic behavior. Interesting to say, a major reduction of this artifact has been achieved by changing the properties of the random variables devoted to round the number of times every rule is applied: by altering their uniform probability so to privilege truncation and discourage rounding toward one, then constant rise-ups of the reactant are almost completely removed.

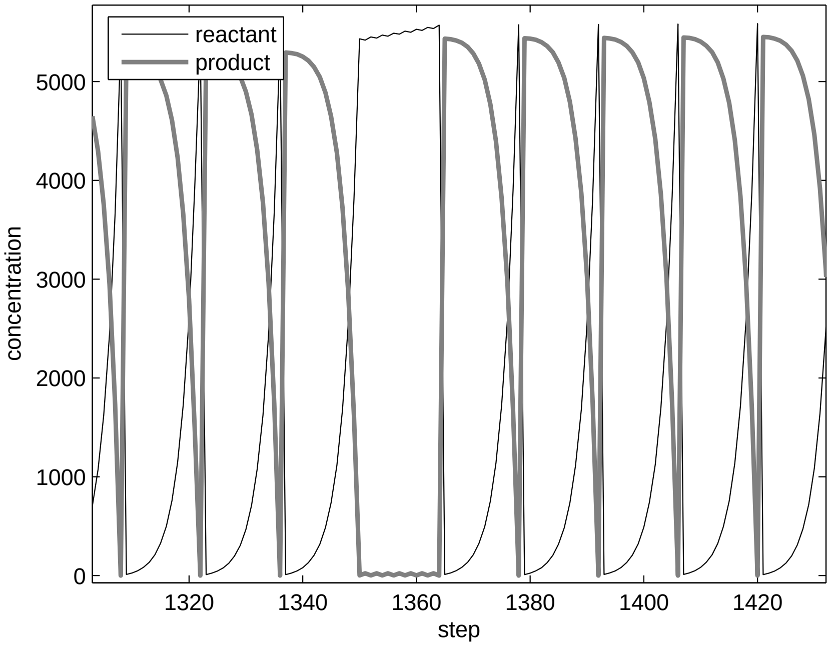

Fig. 3. Brusselator dynamics.

As in the predator-prey model we notice that individual differences in the rule application, occurring when there are few reactant and/or product objects, turn into differences in the system behavior. Though, the BZ reaction is more robust against perturbations and exhibits an asymptotic long-term behavior, that is, individual events affect only the short-term evolution: as opposite to a Lotka-Volterra's population dynamics, in which the evolution of the system entirely depends on its internal state, the Brusselator is, in fact, driven by a constant incoming and dissolving of the reactant, accounted for respectively by r and u. This streaming activity adds inherent robustness to the model of a BZ reaction.

4.3 PKC activation

As another case study, we consider here a simple signal transduction network describing the activation of the protein kinase C (PKC) [12, 13]. The importance of this process is due to the fact that PKC mediates many cellular responses to extracellular stimuli and is involved in several regulatory phosphorylations dealing with proliferation, apoptosis and differentiation. PKC activation is elicited by the allosteric effect of calcium ions (Ca), whose affinity is increased by other agents such as arachidonic acid (AA) and diacylglycerol (DAG).

We refer to the PKC activation model discussed in [2], to which we send the reader for further details. We have translated this model into the following set of rules:

$$
\begin{array}{lll}
r_1 : PKC-i & \rightarrow & PKC-a \\
r_2 : PKC-a & \rightarrow & PKC-i \\
r_3 : PKC-i\ AA & \rightarrow & PKC-aA \\
r_4 : PKC-aA & \rightarrow & PKC-i\ AA \\
r_5 : Ca.PKC & \rightarrow & PKC-aC \\
r_6 : PKC-aC & \rightarrow & Ca.PKC \\
r_7 : Ca.PKC\ AA & \rightarrow & PKC-aCA \\
r_8 : PKC-aCA & \rightarrow & Ca.PKC\ AA \\
r_9 : DAG.Ca.PKC & \rightarrow & PKC-aD \\
r_{10} : PKC-aD & \rightarrow & DAG.Ca.PKC \\
r_{11} : AA.DAG.PKC & \rightarrow & PKC-aAD \\
r_{12} : PKC-aAD & \rightarrow & AA.DAG.PKC \\
r_{13} : PKC-i\ Ca & \rightarrow & Ca.PKC \\
r_{14} : Ca.PKC & \rightarrow & PKC-i\ Ca \\
r_{15} : Ca.PKC\ DAG & \rightarrow & DAG.Ca.PKC \\
r_{16} : DAG.Ca.PKC & \rightarrow & Ca.PKC\ DAG \\
r_{17} : DAG.PKC\ AA & \rightarrow & AA.DAG.PKC \\
r_{18} : AA.DAG.PKC & \rightarrow & DAG.PKC\ AA \\
r_{19} : PKC-i\ DAG & \rightarrow & DAG.PKC \\
r_{20} : DAG.PKC & \rightarrow & PKC-i\ DAG
\end{array}
\tag{30}
$$

in which AA, Ca, and DAG have the meaning introduced previously and we use the symbols $PKC{-}i$ and $PKC{-}a$ to denote respectively the inactivated and activated form of protein kinase C. All remaining symbols represent intermediate complexes. Moreover, for every object X of the system we have introduced a transparent rule of the form $X \rightarrow X$ (not represented in the set of equations above). Note that the rules just expressed represent biochemical reactions mediated by enzymes. For this reason each rule r_i is coupled with a rate constant k_i. The rate constants used in our simulations are taken directly from [2] and are summarized below:

$$
\begin{array}{lll}
k_1 = 50 & k_2 = 1 & k_3 = 0.1 \\
k_4 = 2 \cdot 10^{-10} & k_5 = 3.5026 & k_6 = 1.2705 \\
k_7 = 0.1 & k_8 = 2 \cdot 10^{-9} & k_9 = 0.1 \\
k_{10} = 1 & k_{11} = 0.2 & k_{12} = 2 \\
k_{13} = 0.5 & k_{14} = 1 \cdot 10^{-6} & k_{15} = 8.6348 \\
k_{16} = 1.3333 \cdot 10^{-8} & k_{17} = 2 & k_{18} = 3 \cdot 10^{-8} \\
k_{19} = 0.1 & k_{20} = 1 \cdot 10^{-9} &
\end{array}
\tag{31}
$$

For every rule r_i we define a reactivity map to be simply the corresponding rate k_i:

$$
F_{r_i} = k_i \quad , \quad i = 1, \ldots, 20 \tag{32}
$$

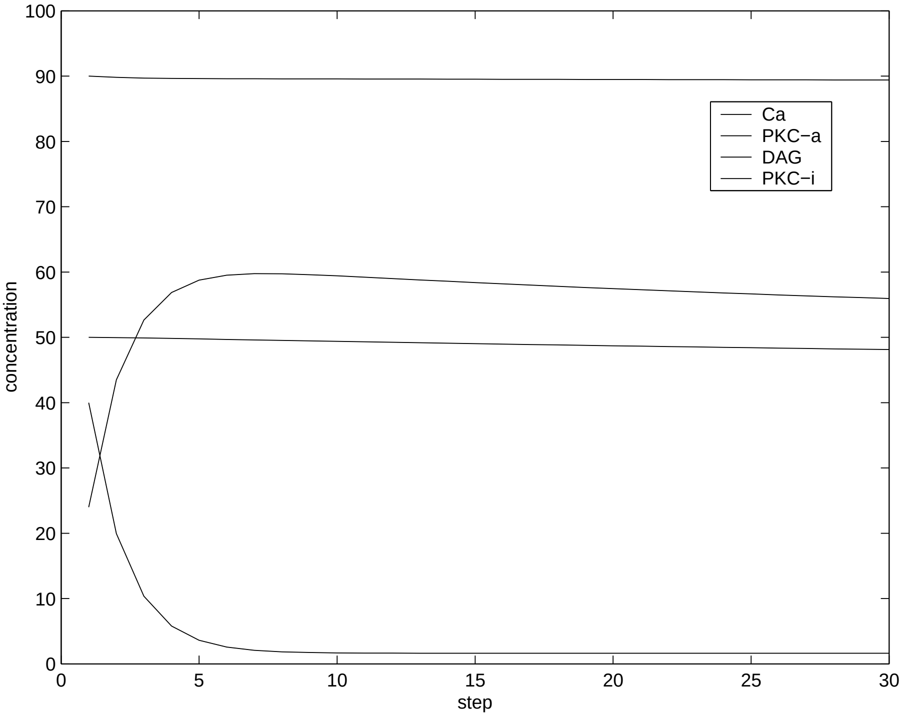

Fig. 4. PKC activation dynamics. The order of the elements in the legend is the same as the order of their final concentrations within the plot.

meanwhile we associate a constant reactivity map to each transparent rule, in our case $F = 50$. These reactivity maps are quite simple but in the future we intend to investigate the effectiveness of more complex reactivity maps in the case of the PKC model.

The description of the whole set of weights W_{r_i}, that can be calculated object by object in the way introduced in previous sections, is omitted. Rather, we present some simulation results obtained using our algorithm. In Figure 4 we see that, in accordance with results obtained in [2], $PKC - i$ decreases to zero while $PKC - a$ grows up until reaching a stationary maximum. Figure 5 represents the characteristic dynamics of the diacylglycerol-protein kinase C (DAG.PKC) complex.

5 Discussion

All rules discussed so far do not present any target specification. This aspect needs further discussion due to its importance.

Let's consider the following rule r, present in a membrane w_i:

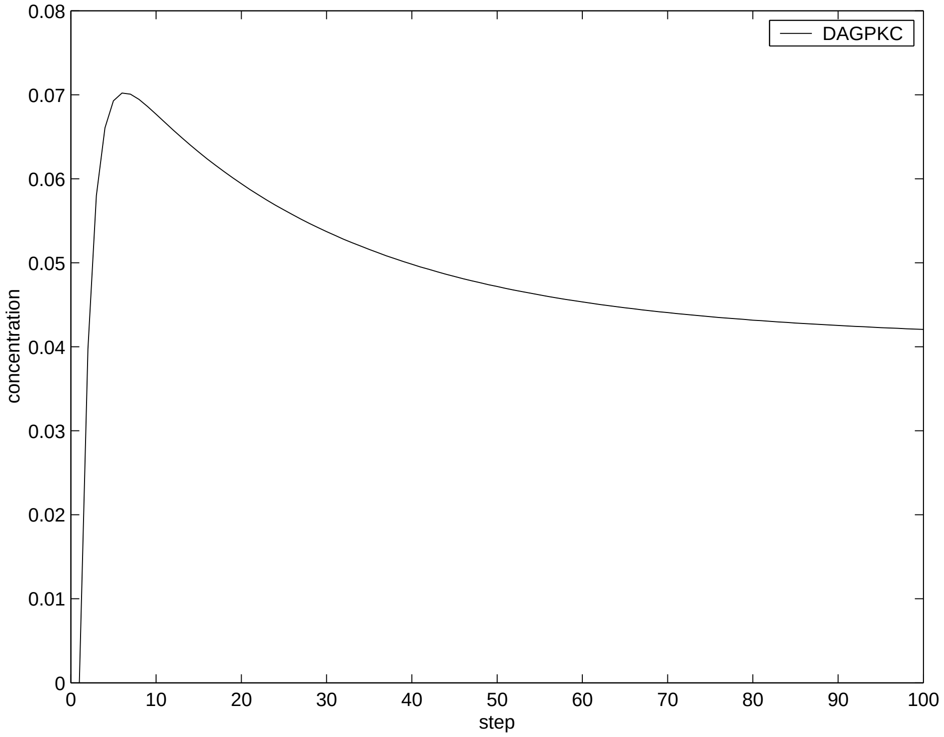

Fig. 5. DAG.PKC complex dynamics.

$$AB \rightarrow B_{IN_j} C \quad , \quad F_r$$

where F_r is the reactivity map associated to r. Its meaning is the following: whenever A joins B inside w_i, they combine and produce an object C inside the same membrane, meanwhile an object B leaves w_i and reaches the membrane w_j. In such a way r affects objects that are present in two different membranes. In particular, from a structural viewpoint, the elements B that are present in w_i have to be distinguished from the elements B that are present in w_j (and, in fact, this is the effect of compartmentalization). For this reason r originates four metabolic equations describing the behavior of its four *distinct* elements:

$$\begin{aligned}\varDelta_r(A_{w_i}) &= -\varLambda_{r,w_i}\\ \varDelta_r(B_{w_i}) &= -\varLambda_{r,w_i}\\ \varDelta_r(B_{w_j}) &= +\varLambda_{r,w_i}\\ \varDelta_r(C_{w_i}) &= +\varLambda_{r,w_i}\end{aligned}$$

where we have introduced the label of the membrane containing every element as subscript.

In this way we can see that the variation due to r on the objects B placed inside w_j, i.e., $\varDelta_r(B_{w_j})$, depends on the concentrations of A and B located in w_i

as stated by the subscript notation (note that this dependence is hidden behind the Λ_r factor). This simple evolution rule is powerful enough to show that movements of objects between membranes can be handled easily by considering, as distinct elements, two objects of the same type located in different regions. This additional information introduces a notational overhead of targeting every object with the label of the membrane containing the respective object. On the other hand it does not introduce any conceptual complication.

The case in which elements appearing in the antecedent of the rule are placed inside different membranes can be handled similarly. The only difference that must be taken into account is that the set of weights has to be calculated by considering, in principle, the whole set of rules of the P system rather than the set of rules of a single membrane.

6 Conclusion and Ongoing Research

Systems biology demands for novel procedures capable of representing biological processes with both accuracy and flexibility. In front of a huge and well-rooted family of numerical schemes, traditionally devoted to figure out the dynamics of systems described by differential equations, alternative algorithms based on a symbolic representation of the phenomena promise to deal more naturally with the structural characteristics of the biomolecules and with the biochemical reactions such molecules give rise to. By using the same kind of representation, our algorithm moreover seems to handle in a straightforward and efficient way those conditions in which few molecules have an important impact on the system dynamics, where most traditional numerical strategies are no longer effective and must be substituted by stochastic algorithms.

Successful simulations conducted on two paradigmatic nonlinear processes in biochemistry, namely the Lotka-Volterra population dynamics and the BZ reaction, plus the experiment conducted with the PKC activation process, ask for further test the potential of the metabolic algorithm. Our present research aims to simulate some fundamental signal transduction networks, in particular the PER and TIM cycle in the circadian oscillation in *Drosophila*.

References

1. D. Besozzi, G. Ciobanu: A P system description of the sodium-potassium pump. In *Membrane Computing, 5th International Workshop, WMC 2004* (G. Mauri, G. Păun, M.J. Pérez–Jiménez, G. Rozenberg, A. Salomaa, eds.), LNCS 3365, Springer-Verlag, Berlin, 2005, 210–223.
2. U.S. Bhalla, R. Iyengar: Emergent properties of networks of biological signaling pathways. *Science*, 283 (January 1999), 381–387.
3. L. Bianco, F. Fontana, G. Franco, V. Manca: P systems for biological dynamics. In [4].

4. G. Ciobanu, Gh. Păun, M.J. Pérez–Jiménez, eds.: *Applications of Membrane Computing.* Springer, Berlin, 2005.
5. I.R. Epstein, K. Showalter: Nonlinear chemical dynamics: Oscillations, patterns, and chaos. *J. Phys. Chem.*, 100, 31 (1996), 13132–13147.
6. A. Goldbeter: Computational approaches to cellular rhythms. *Nature*, 420 (November 2002), 238–244.
7. H. Kitano: Computational systems biology. *Nature*, 420 (November 2002), 206–210.
8. J.C. Leloup, A. Goldbeter: A model for circadian rhythms in *Drosophila* incorporating the formation of a complex between the PER and TIM proteins. *J. of Biological Rhythms*, 13, 1 (1998), 70–87.
9. A. Lindenmayer: Mathematical models for cellular interaction in development. Part I and part II. *J. of Theoretical Biology*, 18 (1968), 280–315.
10. A.J. Lotka. Undamped oscillations derived from the law of mass action. *J. Am. Chem. Soc.*, 42 (1920), 1595–1599.
11. V. Manca, L. Bianco, F. Fontana: Evolutions and oscillations of P systems: Theoretical considerations and applications to biochemical phenomena. In *Membrane Computing, 5th International Workshop, WMC 2004* (G. Mauri, Gh. Păun, M.J. Pérez-Jiménez, G. Rozenberg, A. Salomaa, eds.), LNCS 3365, Springer-Verlag, Berlin, 2005, 63–84.
12. A.C. Newton: Protein Kinase C: Structure, function, and regulation. *J. Biol. Chem.*, 270 (1995), 28495–29498.
13. K. Ohkusu: Elucidation of the protein kinase C-dependent apoptosis pathway in distinct subsets of T lymphocytes in mrl-lpr/lpr mice. *Eur. J. Immunol.*, 25 (995), 3180.
14. M.J. Perez-Jimenez, F.J. Romero-Campero: Modelling EGFR signalling network using continuous membrane systems. *Third Workshop on Computational Methods in Systems Biology*, Edinburgh, 2005.
15. P. Prusinkiewicz, M. Hammel, J. Hanan, R. Mech: Visual models of plant development. In *Handbook of Formal Languages* (G. Rozenberg, A. Salomaa, eds.), volume III: *Beyond Words*, Springer-Verlag, Berlin, 1997, 535–597.
16. Gh. Păun: Computing with membranes. *J. Comput. System Sci.*, 61, 1 (2000), 108–143.
17. C.V. Rao, D.N. Wolf, A.P. Arkin: Control, exploitation and tolerance of intracellular noise. *Nature*, 420 (November 2002), 231–237.
18. G. Rozenberg, A. Salomaa, eds.: *Handbook of Formal Languages.* Springer-Verlag, Berlin, 1997.
19. I. Stamatopoulou, M. Gheorghe, P. Kefalas: Modelling dynamic organization of biology-inspired multi-agent systems with communicating x-machines and population p systems. In *Membrane Computing, 5th International Workshop, WMC 2004* (G. Mauri, Gh. Păun, M.J. Pérez-Jiménez, G. Rozenberg, A. Salomaa, eds.), LNCS 3365, Springer-Verlag, Berlin, 2005, 389–403.
20. Y. Suzuki, H. Tanaka: Chemical oscillation in symbolic chemical systems and its behavioral pattern. In *Proc. International Conference on Complex Systems* (Y. Bar-Yam, ed.), Nashua, NH, September 1997.
21. V. Volterra: Fluctuations in the abundance of a species considered mathematically. *Nature*, 118 (1926), 558–560.