Computing and Informatics, Vol. 27, 2008, 467–479

FREQUENCY MEMBRANE SYSTEMS

Davide Molteni, Claudio Ferretti, Giancarlo Mauri

Dip. di Informatica, Sist. e Com.
Univ. di Milano-Bicocca
Via Bicocca degli Arcimboldi 8
I-20126, Milano, Italy
e-mail: ferretti@disco.unimib.it

Revised manuscript received 3 December 2007

Abstract. We define a model of membrane system where each membrane is clocked independently from the others, in the sense that every derivation step is applied without a global synchronization. The computation is obtained by the execution of a limited amount of rules in each membrane, and only when they are allowed to execute a derivation step. Indeed, each membrane operates with a certain work frequency that can change across the system. Simple results show that this model is at least as powerful as the usual one, and the goal is to present a few examples that show it giving rise to interesting dynamic behaviors.

1 INTRODUCTION: BIOLOGICAL MOTIVATION

This work introduces a new class of P systems, where we want to come closer to biological cell's behavior by adopting some new features of membrane computing that could have a higher correspondence in biology. As we know, in general, different chemical reactions can take different time to be executed and moreover the same reaction could take different time depending on some environment conditions (for example concentration, temperature, etc.).

Some models have already adopted different approaches to the timing in P systems. For instance, in [8] maximal parallelism is not enforced, while in [1] a single rule is probabilistically chosen inside each membrane, but all selected rules work in parallel. Interestingly, paper [2] specifically aims at modeling some biological phoenomena. In [4, 10, 6] the authors study a model where each rule can take a different time to perform its action, and look for systems where even if timing changes results remain the same. Further results along this line are in [5].

Other works considered timing changes inspired by biology. In [3] the authors define a model where bounds can be imposed on the number of membranes and on the numer of rules to be considered active at each step of computation, and as an example of possible application they present the biological mechanism of quorum sensing in *vibrio fisheri*. Paper [7], motivated by biology, defines membrane systems where the degree of parallelism changes with time according to current state of the system.

We want to explore other timing changes inside P systems but, above all, we want to focus the attention on the real asynchronic nature of biological cells. We want to consider every single membrane as a separate domain with a specific clock, where the rules can be applied as often as specified by their membranes' clock frequency. In this behavior, a single derivation step can occur only when the membrane has the right, specified by its own clock and by other constraints, to carry out one or more rules, so each component of the system, regarding the overall computation, is totally independent from the others, hence the system is partially asynchronous.

As we said before, we also want to adopt a different approach to the maximal parallel way to execute the rules. We think maximal parallelism is a very strong assumption, so we want to bound the amount of rules that can be applied in each membrane's derivation step. The motivations for this feature come from the fact that, in a living cell, there is a limited amount of energy, hence a limited amount of reactions can take place in a given time.

In this paper we introduce also a decay time for some symbol objects; we have made this assumption thinking that other reactions (not described here) could be modeled by such feature. For instance, symbols could decay because the cell itself uses them to get energy.

One more detail related to clocking is the offset on the starting time of operations in each membrane. This, and the other modeling features, are discussed by means of examples and simple formal results in the following sections.

2 FREQUENCY P SYSTEMS: DEFINITION

A *frequency P system* is a P system where each rule has a time of execution. This execution time is expressed in clock steps. Each membrane has its own clock, and a limited amount of rules that can be applied in each clock step (or a fixed amount of energy, if we want to associate different levels of energy to the rules). The clock period is a multiple of the unit time of an external observer.

The system will use symbol objects with evolution rules. We denote by N the set of natural numbers.

A Frequency P system with symbol objects of degree $m \geq 1$, is a construct

$$\Pi = (O, D, \mu, \omega_1, \ldots, \omega_m, E, t_D, Cl, R_1, \ldots, R_m, i_O)$$

where:

- O is the alphabet of the objects;

- $D \subseteq O$ is the alphabet of decaying symbols;
- μ is a membrane structure consisting of m membranes labeled with $1, \ldots, m$; each membrane i can be equivalently denoted by μ_i or $[_i]_i$;
- $\omega_i, 1 \leq i \leq m$, specifies the multiset of objects present in the corresponding region i at the beginning of a computation;
- $E \subseteq N^m$ is a set of m numbers indicating the energy value assigned to each membrane at every membrane's clock step, overriding any previous energy level associated to them; that amount bounds the total energy that can be used by rules applied during the next step;
- $t_D \subseteq N^n$ is a set of n numbers indicating the decay time of the n decay symbols in D;
- $Cl \subseteq N^m$ is a set of m numbers indicating the clock value (referred to an external observer) assigned on each membrane;
- $R_i, 1 \leq i \leq m$ are finite sets of evolutionary rules over O associated with regions $1, 2, \ldots, m$ of μ; the rules can be either cooperative or non-cooperative rules of the form, in the latter case, $A \longrightarrow_s^k v$, where A is an object from O and v is a string over $\{a_{here}, a_{out}, a_{in_j} | a \in O, 1 \leq j \leq m\}$ specifying the target where each produced object a will go: stay in the membrane, leave it, or move directly to membrane j; if the target is not specified, then it is intended to be *here*; k is an integer representing the energy to consume to apply the rule. Note that k could be a negative number, in this case we assume that the reaction modeled by the rule produces energy for the cell; when k is not specified we assume that $k = 1$; s in N is the number of clock steps necessary for the rules to act (and produce the objects on the right hand side); when s is not specified we assume that $s = 0$;
- i_O in $\{0, 1, \ldots, m\}$ is the output region (0 for the environment).

A configuration of $\prod$ at a given time t is represented by a string of parentheses (the structure μ) and strings over O (contents of the regions). For instance, a possible configuration of the system at time 0 (the starting time) with an alphabet $O = \{A, B\}$ and a structure $\mu = [_1[_2]_2[_3]_3]_1$ could be:

$$\varsigma(\Pi(0)) = [_1[_2AA]_2[_3BB]_3]_1.$$

Given a string ω representing a configuration at time t, then all the strings obtained from ω by taking any permutation of the strings, representing the contents of the regions, represent the same configuration at time t.

We suppose the existence of an external global clock that ticks at uniform intervals, taken as time unit, starting at time 0; we also suppose that each membrane has its own clock that ticks at uniform intervals, taken as multiple of the observer's time unit. If not explicitly stated otherwise, we assume that each membrane starts at the same time (time 0 of the observer).

In each membrane of the system we have a finite number of objects from alphabet O, a finite set of evolution rules each one with its own costs (in time and energy) and a finite amount of energy. At each time step of the observer's clock, we have membranes in the system that are allowed to execute their rules according to their own clock ticks and membranes that are not allowed to execute their rules (until their next clock ticks). These membranes could receive object symbols from other membranes if they are the target of some rules, but they are not allowed to deal with those objects until their next clock ticks. We identify former membranes as active membranes and the latter as passive membranes (in a given observer's time).

Derivation Mode. When a membrane becomes active we apply as many rules as we can, according to the left hand side of the rules and the energy necessary to carry out the rules. It is important to understand that the rules are similarly applied in maximum parallel manner as in standard P systems, since only one instance of objects on the left hand side of the rule will be consumed and only one instance of the objects on the right hand side of the rule will be produced after a fixed number of time steps specified by the rule. During the steps of execution time of a rule, the occurrences of symbol objects in the right hand side are not yet available for other rules. Each time a rule is applied we decrease the energy value of the membrane by the value specified by the rule, in this way a membrane could execute a fixed number of rules for each clock tick. The energy level of a membrane will be reset to the initial value at the beginning of the next membrane clock tick.

This is similar to *sequential* derivation, as defined in [9], but could be formally defined by using the notation introduced there.

Nondeterminism. If two or more rules in an active membrane are allowed to be executed, then possible conflicts for using the occurrences of symbol objects are solved by assigning the objects in a non-deterministic way.

Halting. The computation halts when at a certain observer's clock step no rule can be applied in any region and there are no rules in execution. The output of a halting computation is the vector of numbers representing the multiplicities of objects present in the output region in the halting configuration.

This is called *total halting* in [1], but also partial halting could be considered.

3 FREQUENCY P SYSTEMS: EXAMPLES

The following example shows how a frequency P system works:

$$\begin{aligned}\Pi_0 \quad = \quad & (O, D, \mu = [_1[_2]_2[_3]_3[_4]_4]_1, \omega_1 = \lambda, \omega_2 = A^5, \omega_3 = B^7, \omega_4 = C^7, \\ & E, t_D, Cl, R_1, R_2, R_3, R_4, i_O = 1),\end{aligned}$$

where:

- $O = \{A, B, C, a, b, c, e\}$; $D = \emptyset$
- $E = \{2, 1, 1, 1\}$
- $t_D(i) = 1$, $\forall i \in D$ (but please note that here D is empty)
- $Cl = \{C_1 = 1, C_2 = 3, C_3 = 2, C_4 = 2\}$
- $R_1 = \{r_1 : abc \to e, r_2 : bc \to a, r_3 : aa \to e\}$
- $R_2 = \{r_4 : A \to a_{out}\}$
- $R_3 = \{r_5 : B \to b_{out}\}$
- $R_4 = \{r_6 : C \to c_{out}\}$

The starting configuration of this frequency P system is represented by Figure 1.

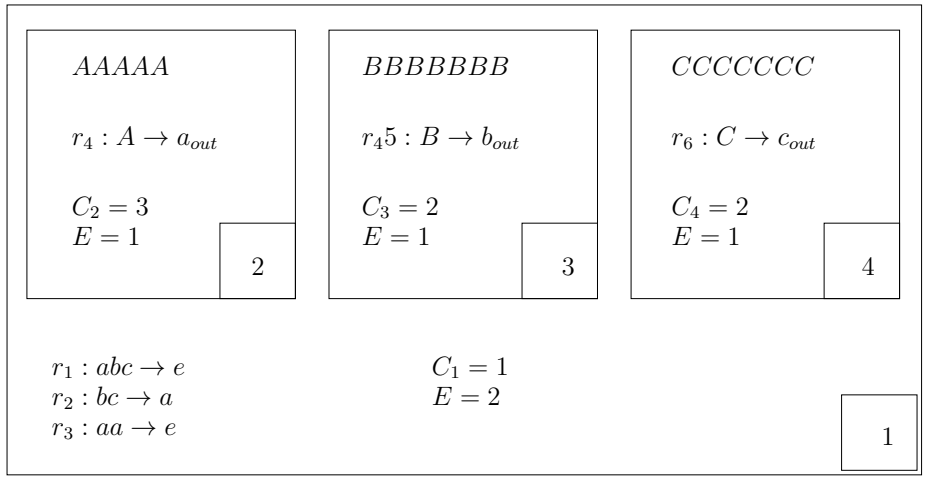

Fig. 1. The starting configuration of Π_0

The configuration of the system at different times is shown in Figure 2. In that figure and in the following ones, we focus the attention only to the objects in skin membrane μ_1, in particular by displaying the availability of symbols a, b, c there at different times, by marking in the picture the axis labeled as "Symbol object...". For instance, in Figure 2 we consider that at time 0 a, b, c are available in skin membrane because produced and moved there by inner membranes in the previous time step, not shown in the figure.

The row labeled μ_1 shows the resulting content of skin membrane after possibly applying rules to the available symbols.

Moreover, consider that since skin mambrane's clock is equal to 1, it coincides with the observer's clock.

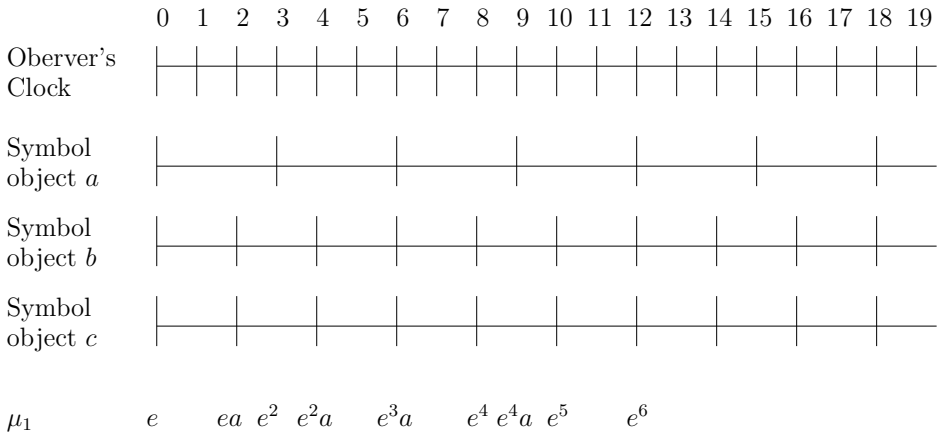

Fig. 2. The evolution of skin membrane in Π_0

Note that, when we set to 0 the time of execution of each rule and each clock is 1, we get the same result of a computation carried out with a standard P system; the only difference is that we introduce a sort of priority in rules execution. For example, if we introduce a new rule in membrane $\mu_1 r_7 : a \to f$) we force the system to apply this rule before the others at least in t5 (because in each clock step the membrane could execute at most 2 rules if it can, and in t5 no other rule can be applied except r_7). In this case we reach the same final configuration of a standard P system also because we set an energy value for the membrane μ_1 to $E = 2$. This parameter in the given example enforces maximality because in case of presence of the symbol objects a, b, c in the membrane we could apply either the rule r_1 or both the rules r_2 and r_3 in the same clock step. This behavior produces the same result (the symbol object e) but in the former case it consumes only a single unit of energy indeed, while in the latter, it consumes two energy units.

Note that this sort of "chain reaction" is allowed by the definitions of the system, even though this happens only because we set to 0 the time of execution of the rules; but, despite of that, this feature can become useful in some further variants of the system (for example we could apply a rule even if the membrane is not active if we have enough energy units left, or we could increase or decrease the clock frequency of the membrane in function of its own energy level).

Note also that if we do not associate decay time to some object symbols we get the same result even if we start the membranes not at the same time.

In the following example we run the same P system as above, but with the difference that we introduce a delay of one observer's clock step for the membrane μ_4. The resulting dynamics is shown in Figure 3.

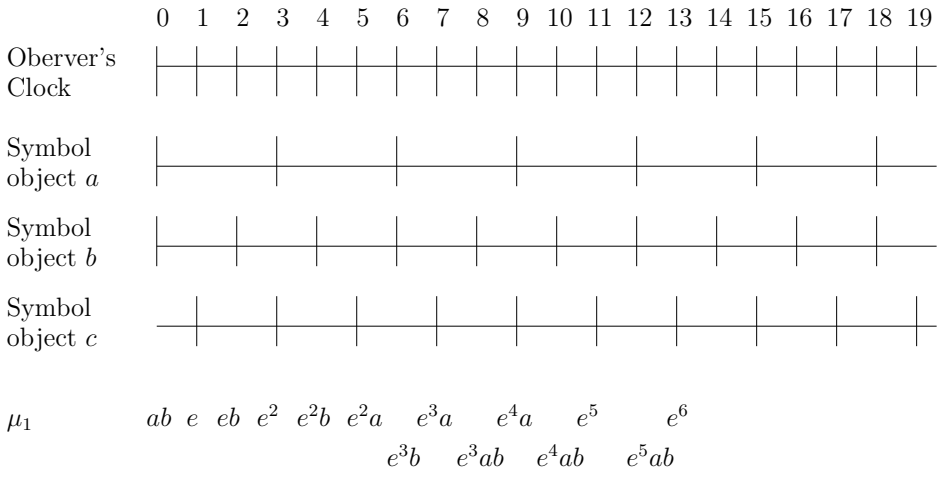

Fig. 3. Dynamics in skin membrane of a system derived from Π_0 by introducing a delay for μ_4

4 FREQUENCY P SYSTEMS WITH DECAY TIME: EXAMPLES

Now we show the same example with a decay time 1 for the symbol objects $\{a, b, c\}$.

The following is the definition of the system Π_1.

$$\begin{aligned}\Pi_1 \quad = \quad & (O, D, \mu = [_1[_2]_2[_3]_3[_4]_4]_1, \omega_1 = \lambda, \omega_2 = A^5, \omega_3 = B^7, \omega_4 = C^7, \\ & E, t_D, Cl, R_1, R_2, R_3, R_4, i_O = 1),\end{aligned}$$

where:

- $O = \{A, B, Cl, a, b, c, e\}$; $D = \{a, b, c\}$
- $E = \{2, 1, 1, 1\}$
- $t_D(i) = 1$, $\forall i \in D$
- $Cl = \{C_1 = 1, C_2 = 3, C_3 = 2, C_4 = 2\}$
- $R_1 = \{r_1 : abc \to e, r_2 : bc \to a, r_3 : aa \to e\}$
- $R_2 = \{r_4 : A \to a_{out}\}$
- $R_3 = \{r_5 : B \to b_{out}\}$
- $R_4 = \{r_6 : C \to c_{out}\}$

As we have done before, the configuration of the system will be shown below. However, first we want to justify this feature by comparing it with its biological counterpart. We could think of an object produced by a membrane as a sort of bio-chemical stimulus that is a part of a reaction. If this stimulus does not reach a certain concentration, the reaction could not take place. We could think this

stimulus will be suppressed if it does not match with a proper receptor in a limited period of time.

When we change the decay time for some symbol objects, the system can have a different behavior (and a different final configuration) depending on the starting time of activity of certain membranes.

In the following examples we have set the decay time $t_D()$ equal to 1 for the symbol objects $\{a, b, c\}$; this means that a symbol object i in D produced at time j, could be used by a rule only from time j (included) to time $j + 1$ (excluded). The dynamics can be seen in Figure 4.

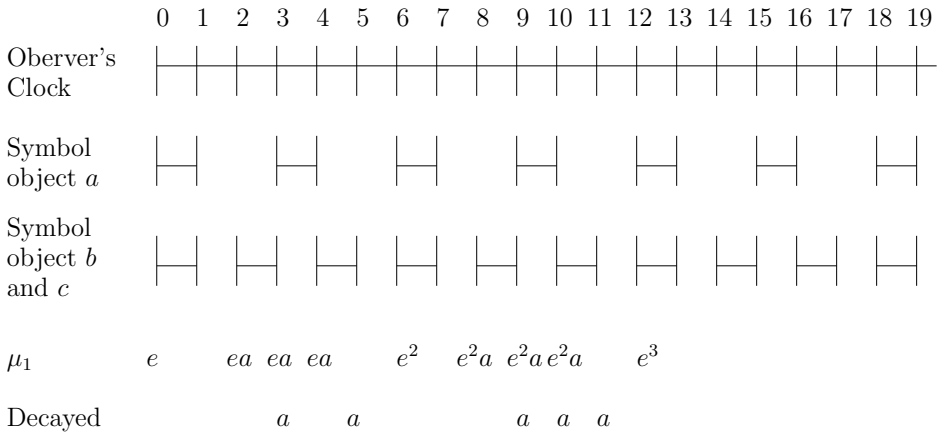

Fig. 4. The dynamics in skin membrane of Π_1

As one can see, in this case the same system with decay times produces a different output with respect to a standard P system and, moreover, the final configuration could be different if we introduce a random delay in membranes starting.

In the previous example we see that the rule r_1 could be applied only where all the three symbol objects $\{a, b, c\}$ are present, and this situation occurs three times: at time t_0, t_6 and t_{12}.

The rule r_2 is applied too, but the symbol object a decays before another instance of the symbol could reach the membrane; hence the rule r_3 is never carried out.

The following example, described by Figure 5, shows the configuration of the same system, but with a delay of one observer's clock step for the membrane μ_4.

Note that no rules of type r_1 or r_3 could be applied during the computation because, due to the time shifting on membrane μ_4, the symbol objects $\{a, b, c\}$ are never present in the membrane at the same time. Hence in this case the halting configuration of the system is $\emptyset$, after consuming all instances of A, B, C.

In the last example we want to force the production of the symbol object e by increasing the reactivity frequency of membrane μ_4. The following is the definition

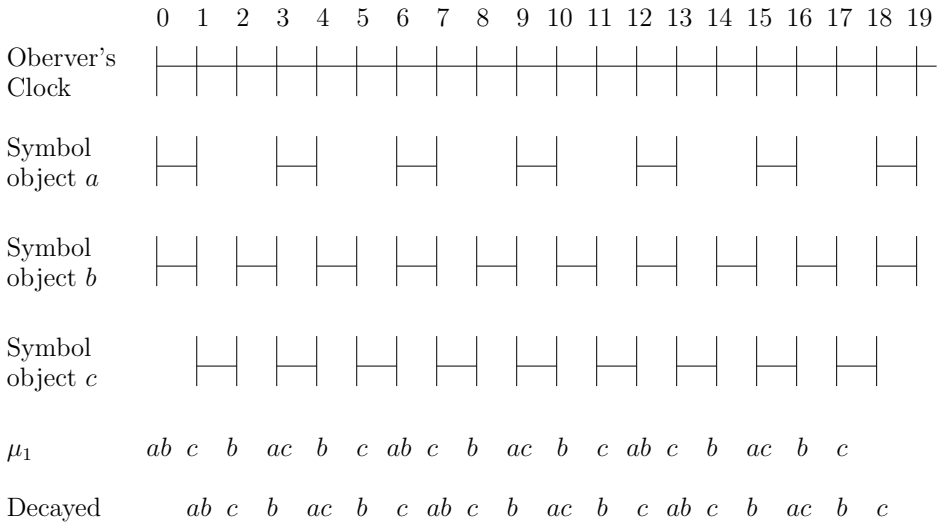

Fig. 5. Dynamics in skin membrane of a system derived from Π_1 by introducing a delay for μ_4

of the system Π_2.

$$\begin{aligned}\Pi_2 \quad = \quad & (O, D, \mu = [_1[_2]_2[_3]_3[_4]_4]_1, \omega_1 = \lambda, \omega_2 = A^5, \omega_3 = B^7, \omega_4 = C^7, \\ & E, t_D, Cl, R_1, R_2, R_3, R_4, i_O = 1),\end{aligned}$$

where:

- $O = \{A, B, Cl, a, b, c, e\}$; $D = \{a, b, c\}$
- $E = \{2, 1, 1, 1\}$
- $t_D(i) = 1$, $\forall i \in D$
- $Cl = \{C_1 = 1, C_2 = 3, C_3 = 2, C_4 = 1\}$
- $R_1 = \{r_1 : abc \to e, r_2 : bc \to a, r_3 : aa \to e\}$
- $R_2 = \{r_4 : A \to a_{out}\}$
- $R_3 = \{r_5 : B \to b_{out}\}$
- $R_4 = \{r_6 : C \to c_{out}\}$

The resulting dynamics is shown in Figure 6.

This behavior is of interest with respect to the modeling of biological mechanisms. We could think that the presence of symbol object e in the halting configuration models certain gene activation. If we simulate several cells with several P systems, derived from Π_1 but with different shifting of their membranes' starting time, we obtain only a little gene activation depending on the starting configuration

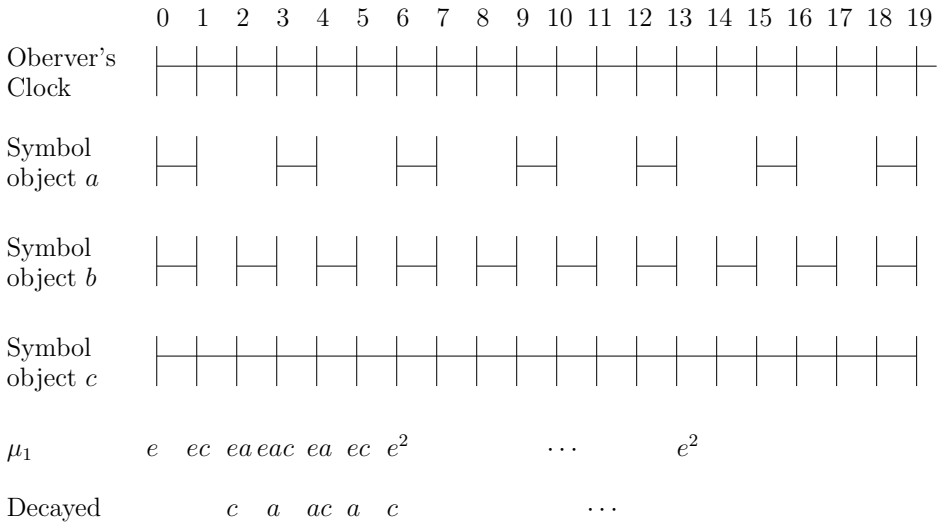

Fig. 6. The evolution of symbols in skin membrane of Π_2

of the membranes of each system, but if we increase the concentration of our stimulus (represented by the frequency increasing of membrane μ_4) we force the activation of the gene in every system for every permutation of their membranes' starting time.

5 SOME PROPERTIES OF FREQUENCY P SYSTEMS

It is easy to state that the class of frequency P systems includes all usual P systems, since we can: assign 0 to the value of energy consumed by each rule, set to 0 the time required by rules to produce their output, have no decaying symbols, have all membranes' clocks with period equal to the unit time of the observer. However, when stating equivalence of models we must check carefully which derivation mode we are considering. All previous examples of frequency membrane systems adopt a derivation mode simlar to the sequential one [8, 9], while usually P systems adopt maximal parallelism. Therefore, the inclusion is true for classes of systems using the same derivation mode.

Another key aspect is that even in our model a global clock does exist. It is the clock of the observer, which has a frequency which is a multple of each membranes' frequency, and above all, all clocks, in membranes and for the observer, are synchronized among themselves, along the grid of time points defined by the observer's clock. Even the eventual offset assigned to membranes is of length equal to an exact multiple of the period of the observer's clock.

We now recall the notion of *partial halting* [1], where computation halts as soon as in at least one mambrane no rule is applicable anymore. This further allows us to state the following result, where we only simulate different frequencies in membranes,

and possibly different length of excution for rules and different starting delays for membranes, in standard P systems:

Proposition 1. Frequency P systems not consuming energy and without decaying symbols, and with skin membrane having period 1 and being the output membrane, can be simulated by usual P systems, when both models use maximal parallelism and partial halting.

Proof. We show that there exists a usual P system simulating a given frequency P system.

We consider observer's clock of simulated frequency P system as the global clock applied in the simulating (usual) P system to each membrane. Membranes of the frequency P system apply rules exactly on (some) ticks of observer's clock. Thus, we can introduce, in the simulating P systems, rules derived from those of the simulated frequency P system, with left side modified by also requiring the presence a new symbol, associated to the correct (cyclic) counting of global clock's ticks: those new symbols are evolved by specific rules so to represent the waiting of simulated frequency membranes.

If the simulated system has membranes with the longest period equal to $p + 1$, we define new "ticking" symbols $t_0, \ldots, t_p$. For instance, for a simulated membrane with period length of 3 we define a simulating membrane where objects t_0, t_1, t_2 will appear, mutually exclusive and in that order, cycling every 3 steps. Each membrane is started having enough symbols t_0, equal in number to the number of the starting objects present in the simulated membrane. The skin membrane, simulating a period of length 1, will not need ticking objects, and their evolution. Instead, it will receive special symbols each time an inner membrane will simulate the application of a rule of the original system, and will destroy them by using special rules.

In terms of halting, the partial halting will detect the stopping of skin membrane, while all other membranes will have to continue to evolve only ticking objects, thus halting the system when the simulated computation halts.

The rules introduced in each inner membrane of the simulating system will have to create correct sequences of ticking objects inside the membrane, and simulate the ones of the frequency system. Moreover, each time a simulating rule acts, it has also to propagate to the outer membrane the special symbol going to tell to skin membrane that the simulated computation is still going on.

Each simulated rule $u \longrightarrow v$ becomes (cooperative): $t_0 u \longrightarrow \ldots v$, in details: if simulated rule takes 0 steps we have $t_0 u \longrightarrow v$, if simulated rule takes 2 steps we have two simulating rules $t_0 u \longrightarrow prj(t_2 2, v), t_2\langle t_2, c\rangle \longrightarrow c$, where $prj(t_2, c) = \langle t_2, c\rangle$, with $\langle t_2, c\rangle$ new symbol. Please note that simulated rules cannot take a number of steps greater than p.

The maximal parallelism, with nondeterminism, will permit to evolve both ticking objects and objects transformed by simulating rules.

Remark: if we also want to simulate a delay in the starting time of a membrane, for instance a delay of 2 steps for a period of length l, it suffices to put in it at the

start, as ticking objects, symbols $t_{(l-2)}$, instead of t_0. Unfortunately, this is not an optimal technique, since it will limit starting delays to $l - 1$. □

Nonetheless, our examples show that the dynamics emerging from the behavior of frequency P systems make them an interesting alternative to usual ones as a way to model some basic mechanisms of biological inspiration.

Acknowledgments

We thank the referees, and we acknowledge the Italian COFIN Project "Modelli di dinamiche cellulari" for partial support.

REFERENCES

[1] Alhazov, A.—Freund, R.—Oswald, M.—Verlan, S.: Partial Versus Total Halting in P Systems. 2006 Brainstorming week on membrane computing, Sevilla, 2007.

[2] Bernardini, F.—Gheorghe, M.—Krasnogor, N.—Muniyandi, R. C.—Perez-Jimenez, M. J.—Romero-Campero, F. J.: On P Systems as a Modelling Tool for Biological Systems. Pre-proceedings of the Sixth Workshop on Membrane Computing (WMC 6), Vienna, June 18–21, 2005, pp. 193–213.

[3] Bernardini, F.—Romero-Campero, F. J.—Gheorghe, M.—Perez-Jimenez, M. J.: A Modeling Approach Based on P Systems with Bounded Parallelism. WMC 7. LNCS 4361, 2006, pp. 49-65.

[4] Cavaliere, M.: Towards Asynchronous P Systems. Pre-proceedings of the Fifth Workshop on Membrane Computing (WMC5), Milano, Italy, June 2004, pp. 12–28.

[5] Cavaliere, M.—Deufemia, V.: Further Results on Time-Free P Systems. Proceedings of the ESF Exploratory Workshop on Cellular Computing (Complexity Aspects), Sevilla (Spain), January 31–February 2, 2005, pp. 95–116, and IJFCS, Vol. 17, 2006, No. 1, pp. 69–90.

[6] Cavaliere, M.—Sburlan, D.: Time and Synchronization in Membrane Systems. Fundamenta Informaticae, Vol. 64, 2005, pp. 1–4, pp. 65–77.

[7] Csuhaj-Varju, E.—Freund, R.—Sburlan, D.: Modeling the Dynamical Parallelism of Bio-Systems. Pre-proceedings of WMC 7, Leiden, 2006, pp. 290–310.

[8] Freund, R.: Asynchronous P-Systems. Pre-proceedings of the Fifth Workshop on Membrane Computing (WMC5), Milano, Italy, June 2004, pp. 12–28.

[9] Freund, R.:—Verlan, S.: A Formal Framework for P Systems. Pre-proceedings of WMC 8, Tessaloniki, 2007, pp. 317–329.

[10] Sburlan, D.: "Clock-Free P Systems. Pre-proceedings of the Fifth Workshop on Membrane Computing (WMC5), Milano, Italy, June 2004, pp. 372–383.

Davide Molteni won a research grant at the University of Milano-Bicocca, after graduating in computer science. His research interests cover computer architectures and simulation of bio-inspired computational models.

Claudio Ferretti is associate professor of computer science at the University of Milano-Bicocca. He received his M. Sc. in computer science from University of Milano in 1988, and the Ph. D. degree in computer science from the Universities of Torino and Milano (associated) in 1993. In 1995 he won a post-doctoral research fellowship, spent in Japan, Tokyo. His main research interests are in theoretical computer science, such as unconventional computing models and formal languages, bio-informatics and molecular computing.

Giancarlo Mauri is full professor of computer science at the Università degli Studi di Milano-Bicocca, Italy, since 1999, where he is director of the Ph. D. program in computer science and member of the Administration Board of the University. Research activity mainly on: formal languages, bio-inspired computing paradigms (DNA computing, P-systems, cellular computing), cellular automata and their applications to the modeling of complex systems, algorithms for sequence analysis in bioinformatics, formal models of systems and processes in biology (systems biology) and their simulation.