Infinite Hierarchies of Conformon-P Systems

Pierluigi Frisco

School of Mathematical and Computer Sciences,
Heriot-Watt University, Edinburgh, EH14 4AS, UK
pier@macs.hw.ac.uk

Abstract. Two models of conformon-P systems, one restricted in the number of input conformons and the other restricted in the number of input membranes, are proved to induce infinite hierarchies.
The described systems do not work under the requirement of maximal parallelism and perform deterministic simulations of restricted counter machines.

1 Introduction

The subdivision of a cell into compartments delimited by membranes has been an inspiration to G. Păun for the definition of a new class of (distributed and parallel) models of computation called *membrane systems* [21].

The hierarchical structure, the locality of interactions, the inherent parallelism, and also the capacity (in the less basic models) for membrane division, represent the distinguishing hallmarks of membrane systems.

Research on membrane systems, also called 'P systems' (where 'P' stands for 'Păun'), has really flourished [22].

One can distinguish three main lines of research concerning membrane systems:

1. establishing their generative power;
2. using them to develop algorithms for solving computationally hard problems;
3. using them as a modelling platform.

In relation with the first item in the previous list an interesting open problem was to find a not universal model of P systems that induces an infinite hierarchy on the number of membranes.

In [12] several models of P systems answering the problem were proposed and accepted as solutions to it. The models described in [12] (and also in subsequent related research [14, 13]) were featured with maximal parallelism, i.e. in each time step the number of performed operations is the maximum number of operations that can be performed.

Here we consider conformon-P systems without maximal parallelism, i.e. in each time step the number of performed operations is any number between 1 and the maximum number of operations that can be performed. We prove that some restricted models of conformon-P systems induce infinite hierarchies on the number of membranes and on the number of input symbols. These results are obtained with deterministic simulations of restricted counter machines.

2 Preliminaries

We assume the reader to have familiarity with basic concepts of formal language theory [11], and in particular with the topic of membrane computing [22]. In this section we recall particular aspects relevant to our presentation.

2.1 Counter Automata

Non-rewriting Turing machines were introduced by M. L. Minsky in [19] and then reconsidered in [20] under the name of *program machines*. After their introduction such machines and some variants of them have been studied under different names: in [9] they were called *(multi)counter machines*, in [1] *multi-pushdown machines*, in [17] *register machines*, and in [10] *counter automata*. Such devices have *counters* (also called registers) each of unbounded capacity recording a natural number or zero.

Simple operations can be performed on the counters: addition of one unit and conditional subtraction of one unit. After each or these operations the machine can change state. The main difference between the original models and some of the subsequent variants indicated above is that the latter may have a read only tape where the input is recorded. In the model introduced by M. L. Minsky, and considered by us, such tape is not present and the input is recorded as a number in one of the counters of the machine. It is shown in [19] that counter automata can simulate any Turing machine (see also [11, Theorem 7.9]).

Formally a counter automaton with n counters ($n \in \mathbb{N}$) is defined as $M = (S, R, s_0, f)$, where S is a finite set of *states*, $s_0, f \in S$ are respectively called the *initial* and *final* state; R is the set of *rules* of the form (s_j, l^+, s_n) (if in state s_j increase the value of the counter l of 1 and go to state s_n), or (s_j, l^-, s_i, s_k) (if in state s_j the value of counter l is zero, then go to state s_k; otherwise decrease the value of the counter l by 1 and then go to state s_i).

Configurations and *computations* for counter automata are defined as in [19].

2.2 Conformon-P systems

In [6], Frisco & Ji introduced a variant of membrane systems called *conformon-P systems* (cP systems). This variant, later studied also in [7, 3–5], is based on simple and basic concepts inspired by a theoretical model of the living cell centred around *conformon* [15, 16].

The concept of conformon was introduced in molecular biology independently in [8] and [25]. The common part of the two definitions is the conformational deformation of (macro) molecules in a cell.

A cP system has conformons, a name-value pair, as objects. If V is an alphabet (a finite set of letters) and $\mathbb{N}_0$ is the set of natural numbers (with 0 included), then we can define a conformon as $[\alpha, a]$, where $\alpha \in V$ and $a \in \mathbb{N}_0$, we will say that α is the *name* and a is the *value* of the conformon $[\alpha, a]$. If, for instance, $V = A, B, C, \dots, Z$, then $[A, 5], [C, 0], [Z, 14]$ are conformons, while $[AB, 21], [C, -15]$, and $[D, 0.5]$ are not.

Two conformons can interact according to an *interaction rule*. An interaction rule is of the form $r : \alpha \overset{n}{\rightarrow} \beta$, where r is the label of the rule, $\alpha, \beta \in V$, and $n \in \mathbb{N}_0$, and it says that a conformon with name α can give n from its value to the value of a conformon having name β. A rule r can be applied only if the value of the conformon with name α is greater than or equal to n. If, for instance, there are conformons $[G, 5]$ and $[R, 9]$ and the rule $r : G \overset{3}{\rightarrow} R$, then the application of r leads to $[G, 2]$ and $[R, 12]$.

The compartments (membranes) present in a cP system have a label, every label being different. Compartments can be unidirectionally connected to each other and for each connection there is a *predicate*. A predicate is an element of the set $\{\geq n, \leq n \mid n \in \mathbb{N}_0\}$. Examples of predicates are: $\geq 5, \leq 2$, etc. If, for instance, there are two compartments (with labels) m_1 and m_2 and there is a connection from m_1 to m_2 having predicate ≥ 4, then conformons having value greater than or equal to 4 can pass from m_1 to m_2. In a time unit any number of conformons can move between two connected membranes as long as the predicate on the connection is satisfied. Notice that we have *unidirectional connections* that is: m_1 connected to m_2 does not imply that m_2 is connected to m_1. Moreover, each connection has its own predicate. If, for instance, m_1 is connected to m_2 and m_2 is connected to m_1, the two connections can have different predicates.

Maximal parallelism, i.e. the fact that in each time step the number of performed operations is the maximum number of operations that can be performed, feature present in most of the variants of P systems, is absent in cP systems. In each time step of a cP systems the number of performed operations is any number between 1 and the maximum number of operations that can be performed.

A computation halts when one (any) conformon is present in a specific (acknowledgement) membrane. When this happens no operation is performed even if it could.

2.3 Some modules for conformon-P systems

In the following we will use the concept of *module*: a group of membranes with conformons and interaction rules in a cP system able to perform a specific task.

An example of module is a *splitter* [3]: a module that, when a conformon $[X, x]$ with $x \in \{x_1, \ldots, x_h\}, x_i < x_{i+1},\ 1 \leq i \leq h-1$ is associated with a specific membrane of it, it may pass such a conformon to other specific membranes according to its value x. In Figures 2 and 3 splitters are depicted by a thicker line, their label starts with **spl**, and their edges have '=' as predicate.

Some of the links between membranes present in the cP systems depicted in Figures 2 and 3 have predicates of the kind $[A, a]$ (a conformon). This is a shorthand for a *separator* module [3]: when conformons of type $[X_i, x],\ 1 \leq i \leq h,\ x \geq 1$ are associated with a specific membrane of it, a separator may pass them to specific different membranes according to their name content. So if there is an edge between membrane 1 and membrane 2 having $[A, a]$ as predicate, it means that only the conformons $[A, a]$ can pass from membrane 1 to membrane 2.

The combination of splitters and separators allows us to define *strict interaction* rule: $A^{(\alpha)} \xrightarrow{\gamma} B_{(\beta)}$ where $\alpha, \beta, \gamma \in \mathbb{N}_0$, meaning that a conformon with name A can interact with B passing just γ only if the value of A and B before the interaction is α and β respectively. The detailed module for strict interaction is depicted in Figure 1. Notice that in a strict interaction just γ is passed even if the value of A could be decreased by any multiple of γ.

Similarly interactions of the kind $A \xrightarrow{\gamma} B_{(\beta)}$ and $A^{(\alpha)} \xrightarrow{\gamma} B$ can be defined.

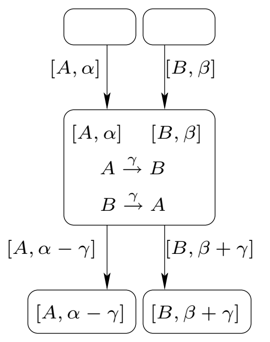

Fig. 1. A detailed strict interaction

In Figures 2 and 3 a forward slash (/) indicates different possibilities for conformons' values, predicates, etc. Moreover, circles with a number indicate membranes having that number as label.

3 Infinite hierarchies

The search for a non-Turing-complete model of P system for which the number of membranes induces an infinite hierarchy on the computation that can be performed by such system was raised in [22]. Moreover, a prize on the description of such system was advertised in [26].

Candidate solutions to this problem were reported in [2, 18], but they were based on definitions that were considered too restrictive, so they were not accepted as solutions.

In [12] several models of P systems answering the problem were proposed and accepted as solutions to it. There *restricted communicating P system* (RCPS) and *restricted counter machines* (RCM) were defined. A RCM is a counter machine which is restricted in its operations: it can increase the value of a counter, say C, only if it decreases the value of another counter, say D at the same time. The counters C and D are said to be *connected*. The research presented in [12] was followed by [14, 13] where infinite hierarchies on the number of symbols or on the number of membranes on the computational power of (restricted) variants of P systems were presented. These studies concerned models of P systems with maximal parallelism.

In the following sections we will describe how two variants of P systems without maximal parallelism, basic conformon-P system with restricted features, can induce infinite hierarchies on the computation they can perform. These results are obtained with a deterministic simulation of RCPS.

With *deterministic simulation* we mean that if the simulated RCM is deterministic, then there will be an isomorphism between the sequence of configurations in the computation of the RCM and some configurations in the basic cP system. This does not mean that in the basic cP system the operations allowing the transition from the simulation of one configuration to the one that deterministically comes after it follow a deterministic path, actually the cP systems defined in the following are non-deterministic. These concepts will be discussed in more details in Section 4.

If the simulated RCM is non-deterministic (one state can be followed by more than one), then in a similar way the simulating basic cP system will be non-deterministic.

3.1 Hierarchy on the number of membranes

In this section we consider *conformon-restricted basic cP systems*, i.e. basic cP systems having a conformon with a distinguished name, let us say l, and such that only some membranes (called *input* membranes) contain only l conformons in the initial configuration (this restrictions is equivalent to the one imposed to the RCPSs presented in [12], where only the object o is used to store the value of the simulated RCM).

Conformon-restricted basic cP systems are accepting devices: a computation is a finite sequence of configurations with the initial configuration having some l conformons in the input membranes (and no conformons in the acknowledgement membrane), while the last (*final*) configuration is the only one in the sequence having one (any) conformon in the acknowledgement membrane. As customary in cP systems, when a final configuration is reached no operation is performed even if it could. If for an input there is such a computation, then we say that the conformon-restricted basic cP system *accepts* the input.

Conformon-restricted basic cP systems are equivalent to RCM.

Lemma 1. *Conformon-restricted cP systems can perform a deterministic simulation of a RCM $M = (S, R, s_0, f)$ with two (connected) counters.*

Proof. First we explain the general idea, then we will go into the details of the proof. Let us assume that M has two counters: c_1 and c_2 whose content is encoded into the value of one conformon l initially present in membrane 5 and another initially present in membrane 4. We will refer to the former of this conformon with $l_{(m5)}$ and with $l_{(m4)}$ to the latter. Moreover here we concentrate only on the operations performed on the value of $l_{(m5)}$ knowing that the value of $l_{(m4)}$ changes in the opposite way (as the two counters are connected). If the value of counter c_1 is 0 (1), then also the value of $l_{(m5)}$ is 0 (1, respectively); if the value of c_1 is $x > 1$, then the value of $l_{(m5)}$ is $2(x-1)+1$. This means that if the value of c_1 is 0 and it is increased by 1, then the value of $l_{(m5)}$ is increased also by 1; if the value of c_1 is bigger than 1 and it is increased by 1, then the value of $l_{(m5)}$ is increased by 2. Similarly for a subtraction: if the value of c_1 is 1 and it is decreased (by 1), then the value of $l_{(m5)}$ (is 1 and it) is decreased by 1; in all the other cases the value of $l_{(m5)}$ is decreased by 2.

The cP system is aware of the value of the c_1 counter through some 'state' conformons (defined later on). If the value of the c_1 counter is 0, then the 'state' conformon will carry this information and no further subtraction will be simulated until an addition is performed on that counter. If the value of the c_1 counter is bigger than 0, then subtractions can be simulated.

In this way the number of 'state' conformons is increased by $|S| \times n \times 2$ (where $|S|$ is the number of states and n is the number of counters of M), but the resulting cP system performs a deterministic simulation (and can be computed in polynomial time).

Now we will explain the cP systems in details; during this proof we will refer to Figures 2 and 3.

The initial configuration of the cP system (identified by the conformons in **bold** in Figures 2 and 3) is: for each state n of the simulated RCM there are conformons $[\hat{s}_n, 0]$ and $[\hat{s}_n^{=0}, 0]$ present in membrane 1, $[s'_n, 0]$ in membrane 8, $[\bar{s}_n^{=0}, 0]$ in membrane 20, $[s_n, 0]$ and $[s_n^{=0}, 0]$ in membrane 25, and $[\ddot{s}_n, 0]$ in membrane 29. All these s conformons are called 'state' conformons as they are associated with the states of M. Membranes 4 and 5 (input membranes) contain the conformon with name l whose value reflects the content of the two counters in M (as indicated above). The remaining of the initial configuration is: $[c, 1]$ in membrane 2, $[\bar{c}, 0]$ in membrane 6, $[z, 6]$ in membrane 10, $[k, 0]$ in membrane 16, $[w, 4]$ in membrane 18, and $[v, 1]$ in membrane 22. The rest of the system will be introduced during the description.

As indicated before, the cP system has two conformons associated with each state of M: s_j in case the value of $l_{(m4)}$ is bigger than 0 and $s_j^{=0}$ otherwise. Let us assume that the value of $l_{(m4)}$ is bigger than 0 and that M is in state j. In this case $[s_j, 30]$ will be present in membrane 1. If $(s_j, l^+, s_n) \in R$, then the value of conformon $l_{(m4)}$ will be increased by 2 (through the conformon c), the conformon $[s_n, 30]$ is generated using part (2 units) of the value of $l_{(m5)}$ (this is because these two l conformons represent connected counters).

For each rule $(s_j, l^+, s_n) \in R$ the instruction $s_j \overset{20}{\rightarrow} \hat{s}_n$ is in membrane 1. In this way the conformons $[\hat{s}_n, 20]$ and $[s_j, 10]$ are generated and they can pass (through spl_1) to membranes 2 and 3 respectively. In membrane 2 $[\hat{s}_n, 20]$ can give (with strict interaction) 2 to the c conformon and then pass to membrane 3 while the c conformon can pass to membrane 5. After passing 2 units to l the c conformon can pass back to membrane 2. In membrane 3 $[\hat{s}_n, 18]$ and $[s_j, 10]$ can interact such that $[s_j, 0]$ and $[\hat{s}_n, 28]$ are generated and they can pass to membrane 25 and 4 respectively. In membrane 4 the value of $\hat{s}_n$ can be increased by 2 (taken from the value of $l_{(m5)}$), when this happens $[\hat{s}_n, 30]$ can pass to membrane 25. Through membranes 25, 27, and spl_3 $[\hat{s}_n, 0]$ and $[s_n, 30]$ are generated and they can pass to membrane 1.

Now we describe how the simulation of one instruction of the kind $(s_j, l^+, s_n) \in R$ is performed when the value of the l counter is 0. In this case $[l, 0]$ can be in membrane 13 and no l conformon is present in membrane 5. We will see later on how this happens, for the moment let us take this as a fact. Let us also assume the conformon $[s_j^{=0}, 30]$ is in membrane 1, this simulates that M is in state j. As the instruction $s_j^{=0} \overset{21}{\rightarrow} \hat{s}_n$ is also present in this membrane, then the conformons $[s_j^{=0}, 9]$ and $[\hat{s}_n^{=0}, 21]$ can be generated and pass to membranes 14 and 6, respectively (through spl_1). In membrane 6 $[\bar{c}, 2]$ and $[\hat{s}_n^{=0}, 19]$ are generated, afterwards they can pass to membrane 13 and 7, respectively. In membrane 13 $[\bar{c}, 2]$ can give 1 to $[l, 0]$, when this happens $[l, 1]$ can pass to membrane 5, while $[\bar{c}, 1]$ to membrane 7. Notice that in this way the value of l has been increased by 1 (and not 2). When $[\hat{s}_n^{=0}, 19]$ and $[\bar{c}, 1]$ are both in membrane 7 they can interact such that $[\hat{s}_n^{=0}, 20]$ and $[\bar{c}, 0]$ are generated and they can pass to membrane 8 and 6 respectively. When $[\hat{s}_n^{=0}, 20]$ is in membrane 8, then the 'state' conformon $[s_n, 30]$ is generated (indicating that the value of the counter is bigger than 0). This process (similar to what we have described) happens between the membranes 1, 4, 8, 14, 15, 24, 25, 27 and spl_2 and at the end of it $[s_n, 30]$ and $[\hat{s}_n^{=0}, 0]$ are in membrane 1.

Now we describe how an instruction of the kind (s_j, l^-, s_i, s_k) is simulated. In case the 'state' conformon is $s_j^{=0}$, then the counter is empty and the next state is s_k. This is simulated by the rules $s_j^{=0} \overset{16}{\rightarrow} \hat{s}_k^{=0}$ present in membrane 1 and allowing $[s_j^{=0}, 14]$ and $[\hat{s}_k^{=0}, 16]$ to be generated. The creation of $[s_k, 30]$ (similar to what described before) is performed through the membranes 25, 32, 33, spl_1, and spl_2.

In case the 'state' conformon is s_j and an instruction of the kind (s_j, l^-, s_i, s_k) is simulated, then two situations are possible. If the value of $l_{(m5)}$ is 1 (indicating that the counter contains 1), then 1 has to be subtracted by $l_{(m5)}$ and the next 'state' conformon has to be $s_i^{=0}$ (indicating that the counter is empty); if the value of $l_{(m5)}$ is bigger than 2 (indicating that the counter contains at least 2), then 2 has to be subtracted by $l_{(m5)}$ and the next 'state' conformon has to be s_i (indicating that the counter is not empty).

If $[s_j, 30]$ is present in membrane 1 and the instruction (s_j, l^-, s_i, s_k) is simulated, then $[s_j, 11]$ and $[\hat{s}_i, 19]$ are generated and they can pass to membranes 23 and 5 respectively. In membrane 5 $\hat{s}_i$ decreases the value of l of 1 (this is always possible) and then $[\hat{s}_i, 20]$ can pass to membrane 16. Here it interacts with $[k, 0]$ so that $[k, 11]$ and $[\hat{s}_i, 9]$ are created and they can pass to membranes 5 and 17 respectively. The role of $[k, 11]$ is to subtract 1 from the value of l. If this is possible then $[k, 12]$ will also pass to membrane 17, if this is not possible then the k conformon will stay in membrane 5 until the z conformon will be also there.

In case after the interaction with $\hat{s}_i$ the value of l becomes 0, then it can pass to membrane 10 and here interact with $[z, 6]$. As a result of this and other interactions (involving membranes 11 and 12) $[l, 0]$ (originally in membrane 4) can pass to membrane 13 (this fact was considered above), while $[z, 6]$ can pass to membrane 17. It should be clear now that in membrane 17 either $[k, 12]$ or $[z, 6]$ is present, they cannot be present together.

If $[k, 12]$ is present, then the 2 units taken from $l_{(m5)}$ are passed to $l_{(m4)}$ and $[s_i, 30]$ is generated and it can pass to membrane 1; if instead $[z, 6]$ is present in membrane 17, then the unit taken from $l_{(m5)}$ is passed to $l_{(m4)}$ and $[s_i^{=0}, 30]$ is generated and it can pass to membrane 1. These processes, similar to others described above, happen in between the membranes 17-33 (with the exception of membrane 26).

If j is a final state for M, then either $[s_j, 30]$ or $[s_j^{=0}, 30]$ will be present in membrane 1. When this happens, then the application of either $s_j \xrightarrow{18} \hat{s}_f$ or $s_j^{=0} \xrightarrow{18} \hat{s}_f$ will create a conformon with value 18. This conformon can pass to membrane 26 (the acknowledgement membrane) halting in this way the computation. □

The just given constructive proof can be used to create conformon-restricted cP systems that can perform a deterministic simulation of RCMs (with any number of connected counters).

Let us assume that a specific RCM has m counters $C = \{c_1, \ldots, c_m\}$ each with an initial value. Then it is possible to build a conformon-restricted cP systems Π' having l conformons present in m different input membranes. Considering the proof of Lemma 1 this seems to be a must as collecting more than one conformon with name l in the same membrane would not allow the system Π' to perform a simulation on the RCM (we will discuss this point in Section 4). The system Π' would be such that every time the value of an l conformon is increased (decreased), then the one of its connected counter (for the particular simulated instruction) is decreased (increased, respectively) by the same amount. The information on the connected counter can be present in the name or in the value of the 'state' conformons (similarly to what was done in the previous proof).

So we can state:

Corollary 1. *Conformon-restricted cP systems can perform a deterministic simulation of RCMs.*

Here is the reverse of this inclusion:

Lemma 2. *A RCM with m counters can simulate a conformon-restricted cP system having m input membranes.*

Proof. In the initial configuration the value stored in the m counters is the value of the l conformons present in the initial configuration of the cP system. The rest of the description of the cP system is encoded into the finite control of the RCM.

The increase/decrease of the value of the l conformons is split into a sequence of increases/decreases of 1. Every time 1 is subtracted by the value of an l conformon, then the value of the associated counter is decreased by 1 and the one of the connected counter is increased by 1. Chains of coupled increase/decrease of connected counters simulate the passage of value between different conformons. Similarly when the value of an l conformon is increased.

When the simulation of a conformon passing to the acknowledgement membrane is simulated, then the RCM goes into a final state. □

The fact that RCMs induce an infinite hierarchy on the number of counters is proved in [12]. Considering this, we can state:

Theorem 1. *Conformon-restricted cP systems induce an infinite hierarchy on the number of membranes.*

3.2 Hierarchy on the number of symbols

In this section we consider *membrane-restricted basic cP systems*, i.e., basic cP systems in which the number of input membranes is restricted to one and the set of names of input conformons is bounded.

An initial configuration is such that some input conformons are present in the input membrane, no input conformon is present in the remaining membranes, and the acknowledgement membrane is empty. We say that a membrane-restricted basic cP systems *accepts* an input if there is a finite sequence of configurations starting from an initial configuration and ending with a (*final*) configuration (being the last one in the sequence) in which one (any) conformon is in the acknowledgement membrane. As customary in cP systems, when a final configuration is reached no operation is performed even if it could.

Lemma 3. *Membrane-restricted cP systems with two input conformons can perform a deterministic simulation of a RCM $M = (S, R, s_0, f)$ with two connected counters.*

Proof. (sketch) The proof follows the one of Lemma 1. Let us assume that the names of the input conformons are l_1 and l_2 and that their initial value reflects the initial content of the counters in M (in the same way as it is done in the proof of Lemma 1).

The operations performed on l_1 and l_2 are similar to the ones performed to the l conformons in the proof of Lemma 1.

The simulation of an instruction of the RCM changing its state into a final one lets a conformon to go into the acknowledgement membrane halting in this way the computation. □

Let us assume that a specific RCM has m counters $C = \{c_1, \ldots, c_m\}$, then it is possible to build a membrane-restricted cP system having input conformons with name $c_1, \dots, c_m$ in the input membrane. The equivalence between the number of counters and the number of different names of input conformons seems to be a must (we will discuss this point in Section 4). The membrane-restricted cP system would be such that every time the value of an input conformon is increased (decreased), then the one of its connected counter (for the particular simulated instruction) is decreased (increased, respectively) by the same amount. The information on the connected counter can be present in the name or in the value of the 'state' conformons.

So we can say:

Corollary 2. *Membrane-restricted cP systems can perform a deterministic simulation of RCMs.*

Here is the reverse of this inclusion whose proof follows the one of Lemma 2:

Lemma 4. *A RCM with m counters can simulate a membrane-restricted cP system having m input conformons.*

If we now consider that RCMs induce an infinite hierarchy on the number of counters [12], then we have:

Theorem 2. *Membrane-restricted cP systems induce an infinite hierarchy on the number of input conformons.*

4 Final remarks

At the beginning of Section 3 we defined *deterministic simulation* and we indicated that the restricted cP systems used by us are non-deterministic even if they can perform deterministic simulations.

The non-determinism present in the considered cP systems arises from the fact that some operations can be performed in parallel. If, for instance, the operations A and B can be performed in parallel in a cP system, then (as maximal parallelism is not present) A can be applied before B, or B can be applied before A, or A and B can be applied at the same time.

We are going to investigate if it is possible to have the results presented in this paper with deterministic cP systems. In this respect we will consider to use Petri nets [23] as done in [4].

It is also relevant to notice that the use of separator modules on the the conformons representing counters let the sum of these conformons not to be constant but to fluctuate. This fluctuation is due to the separator module used by (variants of) strict interactions (see Figure 1 and [3, Figure 2]). Because of the way the cP systems described in this paper have been devised, these fluctuations do not interfere with the simulations performed by them.

An essential element in the proof of Lemma 1 is the presence of connected loops. Here with *loop* we mean that, during the simulation, some conformons cycle in between some membranes. In the proof of Lemma 1 the k conformon loops between membranes 16, 5, and 17 or between membranes 16, and $(5, 9)^+$ (the superscript $^+$ indicates that k can keep passing between membranes 5 and 9), while the z conformon loops between membranes 10, 11, 12, 17, 5, 9, and 19.

Two loops are *connected* if one can be completed only if the other is taking place. In the proof of Lemma 1, for instance, the k conformon can complete the 16, $(5, 9)^+$ loop only when the z conformon is traversing its loop.

We think that loops are necessary in a system lacking maximal parallelism in order to have some computational power. We also think that the number of loops and their connections can be a measure of computational complexity of system lacking maximal parallelism. The concepts of (connected) loops seems to us to be related to the ones of concurrent steps and subsystem in Petri nets [23]. Moreover, this reminds us of the promoter/repressor mechanism present in cells during protein synthesis [24, chapter 29]. We count to investigate this further.

Deterministic simulation can be also used to obtain other results. In [3] Theorem 3 states that a cP system with unbounded total value can perform a (non-deterministic) simulation of a program machine (meaning that in that proof 0-gamble - see [4] - is present). Considering the proof of Lemma 1 we can then state:

Theorem 3. *Conformon-P systems with unbounded value can perform a deterministic simulation of program machines.*

We conclude this paper posing a problem related to variants of P systems inducing an infinite hierarchy. The proofs on the presence of such hierarchies are based on a mapping from the space of configurations of the simulated system (for instance, an RCM) to the considered P system model. In this mapping the content of the counters of the RCM is represented, for instance, by the number of objects in the P system.

Is it possible to have another mapping such that the infinite hierarchy is not induced?

In Section 3.1 we wrote: "... this seems to be a must as collecting more than one conformon with name l in the same membrane would not allow the system Π' to perform a simulation on the RCM.".

This 'must' has no proof (we were not able to give one), even if our common sense suggests that it is not possible to do otherwise.

References

1. B. S. Baker and R. V. Book. Reversal-bounded multipushdown machines. *Journal of Computer and System Science*, 8:315–332, 1974.
2. R. Freund. Special variants of P systems inducing an infinite hierarchy with respect to the number of membranes. *Bulletin of EATCS*, 75:209–219, 2001.
3. P. Frisco. The conformon-P system: A molecular and cell biology-inspired computability model. *Theoretical Computer Science*, 312(2-3):295–319, 2004.

4. P. Frisco. P systems, Petri nets, and Program machines. In R. Freund, G. Lojka, M. Oswald, and G. Păun, editors, *Proceedings 6th International Workshop on Membrane Computing (WMC6)*, volume 3850 of *Lecture Notes in Computer Science*, pages 209–223. Springer-Verlag, Berlin, Heidelberg, New York, 2006.
5. P. Frisco and R. T. Gibson. A simulator and an evolution program for conformon-P systems. In *SYNASC 2005, 7th International Symposium on Simbolic and Numeric Algorithms for Scientific Computing*, pages 427–430. IEEE Computer Society, 2005. Workshop on Theory and Applications of P Systems, TAPS, Timisoara (Romania), September 26-27, 2005.
6. P. Frisco and S. Ji. Conformons-P systems. In M. Hagiya and A. Ohuchi, editors, *DNA8, 8th International Meeting on DNA Based Computers, Hokkaido University, Sapporo, Japan, June 10-13*, volume 2568 of *Lecture Notes in Computer Science*, pages 291–301, 2002.
7. P. Frisco and S. Ji. Towards a hierarchy of info-energy P systems. volume 2597 of *Lecture Notes in Computer Science*, pages 302–318. Springer-Verlag, Berlin, Heidelberg, New York, 2002.
8. D. E. Green and S. Ji. The electromechanical model of mitochondrial structure and function. *Molecular Basis of Electron Transport*, pages 1–44, 1972. J. Schultz, B. F. Cameron (eds).
9. S. A. Greibach. Remarks on blind and partially blind one-way multicounter machines. *Theoretical Computer Science*, 7:311–324, 1978.
10. H. J. Hoogeboom. Carriers and counters. P-systems with carriers vs. (blind) counter automata. In G. Goos, J. Hartmanis, and J. van Leeuwen, editors, *Developments in language theory, 6th International conference, DLT 2002, Kyoto, Japan, September 2002, Revised papers*, volume 2450 of *Lecture Notes in Computer Science*, pages 140–151, 2003.
11. J. E. Hopcroft and D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.
12. O. H. Ibarra. On membrane hierarchy in P systems. *Theoretical Computer Science*, 334:115–129, 2005.
13. O. H. Ibarra and G. Păun. Characterizations of context-sensitive languages and other languages classes in terms of symport/antiport P systems. *Theoretical Computer Science*, 2006. in press.
14. O. H. Ibarra and S. Woodworth. On bounded symport/antiport P systems. In *Preproceedings of The 11th International Meeting on DNA Computing*, pages 25–36, 2005. London, Ontario, Canada.
15. S. Ji. The Bhopalator: a molecular model of the living cell based on the concepts of conformons and dissipative structures. *Journal of Theoretical Biology*, 116:395–426, 1985.
16. S. Ji. The Bhopalator: an information/energy dual model of the living cell (II). *Fundamenta Informaticae*, 49(1-3):147–165, 2002.
17. J. P. Jones and Y. V. Matijasevič. Register machine proof of the theorem on exponential Diophantine representation of enumerable sets. *Journal of Symbolic Logic*, 49(3):818–829, September 1984.
18. S. Krishna. *Languages of P systems: computability and complexity.* PhD thesis, Indian instituto of technology Madras, 2001. India.
19. M. L. Minsky. Recursive unsolvability of Post's problem of "tag" and other topics in theory of Turing machines. *Annals of Mathematics*, 74(3):437–455, 1961.
20. M. L. Minsky. *Computation: Finite and Infinite Machines.* Automatic computation. Prentice-Hall, 1967.

21. G. Păun. Computing with membranes. *Journal of Computer and System Sciences*, 1(61):108–143, 2000.
22. G. Păun. *Membrane Computing. An Introduction.* Springer-Verlag, Berlin, Heidelberg, New York, 2002.
23. W. Reisig and G. Rozenberg, editors. *Lectures on Petri Nets I: Basic Models*, volume 1491 of *Lecture Notes in Computer Science.* Springer-Verlag, Berlin, Heidelberg, New York, 1998.
24. D. Voet and J. G. Voet. *Biochemistry.* John Wiley and Sons, third edition, December 2003.
25. M. V. Volkenstein. The conformon. *Journal of Theoretical Biology*, 34:193–195, 1972.
26. C. Zandron. P-systems web page: http://psystems.disco.unimib.it.

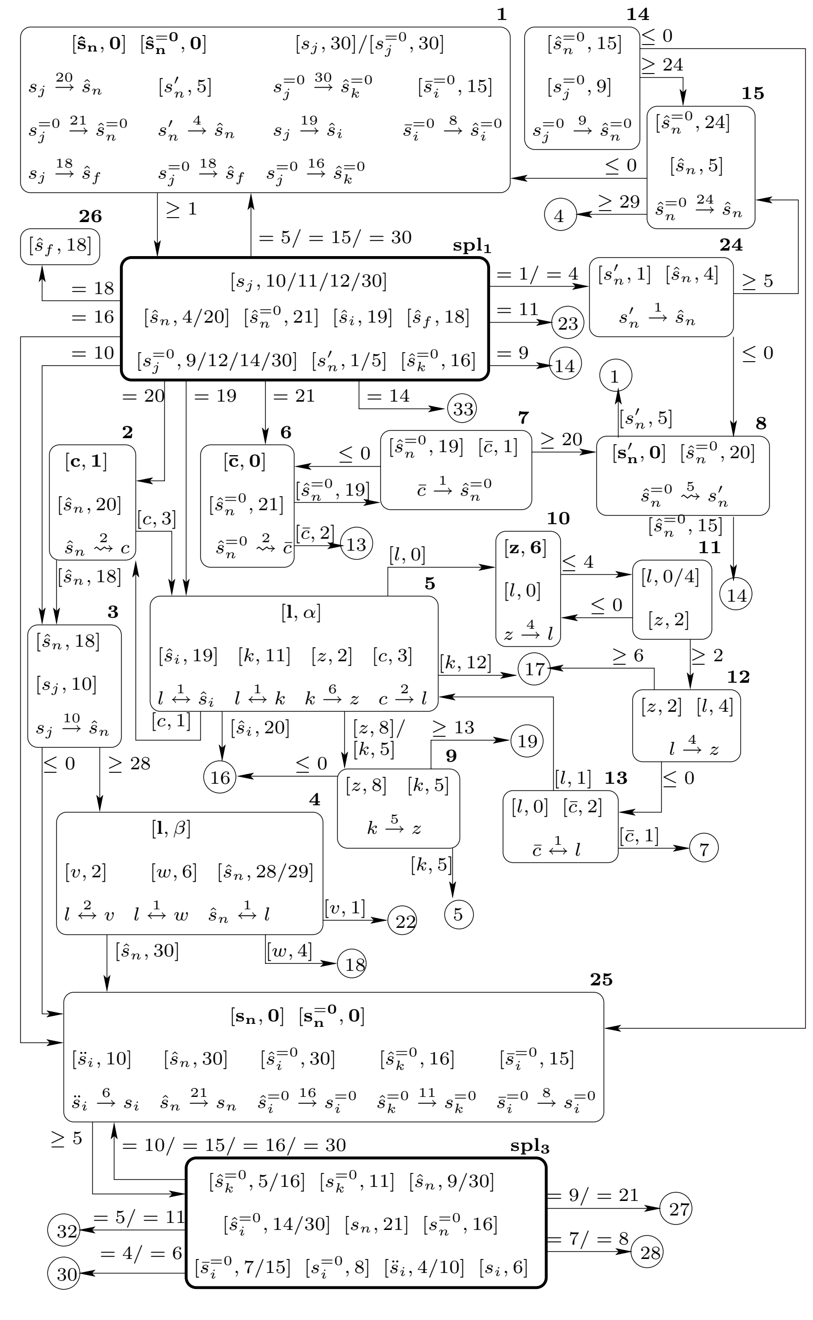

Fig. 2. Conformon-restricted cP systems associate to Lemma 1 (part 1)

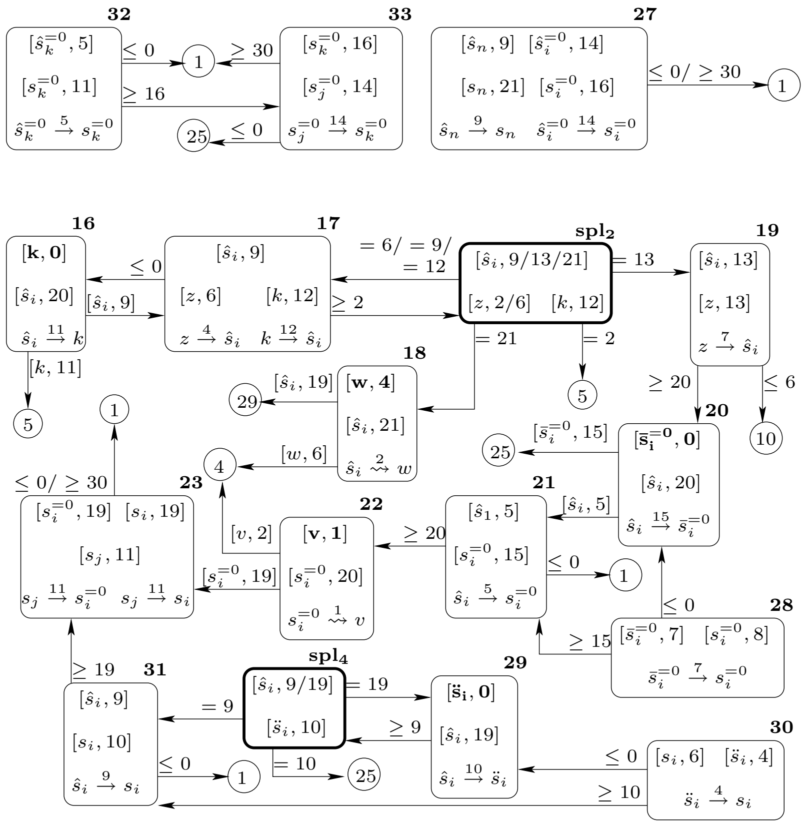

Fig. 3. Conformon-restricted cP systems associate to Lemma 1 (part 2)