No Cycles in Compartments. Starting from Conformon-P Systems

Pierluigi Frisco[1], Gheorghe Păun[2]

[1] School of Mathematical and Computer Sciences
Heriot-Watt University
Edinburgh, EH14 4AS, UK
E-mail: pier@macs.hw.ac.uk

[2] Institute of Mathematics of the Romanian Academy
PO Box 1-764, 014700 Bucharest, Romania
and
Department of Computer Science and Artificial Intelligence
University of Sevilla
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
E-mails: george.paun@imar.ro, gpaun@us.es

Summary. Starting from proofs of results about the computing power of conformon-P systems, we infer several results about the power of certain classes of tissue-like P systems with (cooperative) rewriting rules used in an asynchronous way, without cycles in compartments. This last feature is related to an important restriction appearing when dealing with lab implementations of P systems, that of avoiding local evolution loops of objects.

1 Introduction

This note addresses a technical issue which appeared in the framework of the recent attempt to implement a P system in biochemical terms, at Technion institute, Haifa, Israel, namely of avoiding cyclical evolution of chemicals in any compartment of the system – see a more precise description of the problem in [9]. Here we consider a class of tissue-like P systems, namely as introduced in [16], with rewriting rules present in membranes, and with target indications of the forms *here, go* associated with the "products of reactions": rules of the form $u \to v$, where u and v are multisets of objects and the objects in v have associated target indications *here, go* (actually, *here* is omitted) indicating that the respective object remains in the same compartment or it has to go to any of the adjacent compartments, non-deterministically choosing the destination. We also consider an evolution-communication (EC) version of these systems, following the ideas of [1], i.e., using evolution rules without target indications and using separate communication rules (of the form (a, go), with the obvious meaning: object

a is communicated to any of the adjacent membranes). In order to transfer in a direct way to these systems results from conformon-P systems area, we add to the definition in [16] several "non-standard" ingredients: we work asynchronously (in any step, in any compartment, a rule may be used or not), maybe with a priority relation among rules, of a global type (in each compartment, evolution rules have priority over communication rules: if an object can evolve and, at the same time, communicated, an evolution rule is applied first), an acknowledging membrane (the computation stops when any object is sent to this membrane, which is empty in the beginning of the computation). The number of membranes we use is arbitrary (rather high, if we take into account the number of membranes used in conformon-P systems simulating register machines), but, on the good side, the evolution rules we need to simulate a conformon-P system are of a very restrictive form: each of the multisets u, v from a rule $u \to v$ has exactly two objects.

Although, for the sake of readability, we recall here the definitions of conformon-P systems and of P systems with a graph structure, we do not enter into details, and we assume the reader to be familiar with basic elements of membrane computing. However, we indicate a series of papers related to conformons. This concept was introduced independently in [10] and [17]. Following the definition given in [10] conformons and conformon-like entities have been classified into 10 families according to their biological functions [12]. To know more about the Bhopalator refer to [11, 13]. The term *conformon* was adopted in [14, 15] where the authors started to develop a quantum mechanical theory based on this concept. Conformon-P systems have been introduced in [3] and later studied, among others, in [4, 6]. Conformon-P systems have also been successfully used as a platform to model biological process. The interested reader can refer to [8, 2, 7].

2 Basic Definitions

Let V be an alphabet (a finite set of abstract symbols), and $\mathbb{N}$ be the set of natural numbers, including 0. A multiset over V is a function $M : V \longrightarrow \mathbb{N} \cup \{+\infty\}$. The support of M (the set of elements $a \in V$ for which $M(a) > 0$) is denoted by $supp(M)$ and the cardinality of M (the sum of multiplicities of all elements in $supp(M)$) is denoted by $|M|$.

2.1 Conformon-P Systems

In what follows, a *conformon* is an element of $V \times \mathbb{N}$, denoted by $[a, n]$. We refer to a as the *name* of the conformon $[a, n]$ and to n as its *value*.

Two conformons can interact according to an *interaction rule*. An interaction rule is of the form $a \xrightarrow{e} b$, where $a, b \in V$ and $e \in \mathbb{N}$, and it says that a conformon with name a can give e from its value to the value of a conformon having name b. If, for instance, there are conformons $[a, 5]$ and $[b, 9]$ and the rule $a \xrightarrow{3} b$, one application this rule leads to $[a, 2]$ and $[b, 12]$. As here we consider that the value

of a conformon cannot be a negative number, the rule $a \stackrel{3}{\rightarrow} b$ cannot be applied to $[a, 2]$.

Each membrane present in a conformon-P system has associated a label, different from the labels of other membranes. These membranes are placed in the nodes of a directed graph, hence they are connected in a unidirectionally way. Each connection has associated a *predicate*, which is an element of the set $pred(\mathbb{N}) = \{\geq n, \leq n \mid n \in \mathbb{N}\}$. If, for instance, there are two compartments (with labels) m_1 and m_2 and there is an 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.

A *conformon-P system* is a construct

$$\Pi = (V, \mu, \omega_z, ack, L_1, \ldots, L_m, R_1, \ldots, R_m),$$

where:

V is a finite alphabet;
$\mu = (Q, E)$ is a *directed labelled graph* underlying Π, where
  $Q = \{1, \ldots, m\}$ is the set of *membranes* (we also say *compartments*) of Π;
  $E \subseteq Q \times Q \times pred(\mathbb{N})$ defines directed labelled *edges* between vertices, indicated by $(i, j, pred)$, $i, j \in Q, i \neq j$, where $pred \in pred(\mathbb{N})$ is a *predicate*;
ω_z with $\omega \in \{in, out\}$ and $z \in Q$ indicates whether Π is an accepting ($\omega = in$) or generating ($\omega = out$) device; the compartment z contains the input or output, respectively;
$ack \in Q$ indicates the *acknowledging* compartment;
$L_i : (V \times \mathbb{N}) \rightarrow \mathbb{N} \cup \{+\infty\}$, $i \in Q$, are multisets of conformons initially associated with the vertices in Q;
R_i, $i \in Q$, are finite sets of interaction rules associated with the vertices in Q, with $supp(L_{ack}) = \emptyset$.

Let M_i and R_i be the multiset of conformons and the set of rules, respectively, associated with the compartment $i \in Q$. Two conformons present in compartment i can interact according to a rule in R_i such that the multiset of conformons M_i changes into M_i'. If, for instance, $[a, p], [b, q] \in M_i$, $a \stackrel{e}{\rightarrow} b \in R_i$ and $p \geq e$, then $M_i' = (M_i - \{[a, p], [b, q]\}) \cup \{[a, p - e], [b, q + e]\}$.

A conformon $[a, p]$ present in compartment i can *pass* to compartment j if $(i, j, pred) \in E$ and $pred(p)$ holds. This passage changes the multisets of conformons M_i and M_j into M_i' and M_j', respectively, such that $M_i' = M_i - \{[a, p]\}$ and $M_j' = M_j \cup \{[a, p]\}$.

At the moment we do not assume any requirement (such as maximal parallelism, priorities, etc.) on the application of operations. If a conformon can pass to another compartment or interact with another conformon according to an interaction rule, then one of the two operations or none of them is non-deterministically chosen.

The possibility to carry out one of the two allowed operations in a compartment or none of them lets conformon-P systems to be non-deterministic. Non-determinism can also arise from the configurations of a conformon-P system if in a compartment a conformon can interact with more than one conformon and also from the graph underlying Π if a compartment has edges with the same predicate going to different compartments.

A *configuration* of Π is an m-tuple $(M_1, \ldots, M_m)$ of multisets over $V \times \mathbb{N}$. The m-tuple $(L_1, \ldots, L_m)$, is called *initial configuration* (remember that $supp(L_{ack}) = \emptyset$, so in the initial configuration the acknowledging compartment does not contain any conformon) while any configuration having $supp(M_{ack}) \neq \emptyset$ is called *final configuration*. In a final configuration no operation is performed even if it could.

For two configurations $(M_1, \ldots, M_m), (M'_1, \ldots, M'_m)$ of Π we write $(M_1, \ldots, M_m) \Rightarrow (M'_1, \ldots, M'_m)$ indicating a *transition* from $(M_1, \ldots, M_m)$ to $(M'_1, \ldots, M'_m)$, that is, the application of one operation to at least one conformon. In other words, in any configuration in which $supp(L_{ack}) = \emptyset$ any conformon present in a compartment can either interact with another conformon present in the same compartment or pass to another compartment or remain in the same compartment unchanged. If no operation is applied to a multiset M_i, then $M'_i = M_i$. The reflexive and transitive closure of $\Rightarrow$ is indicated by $\Rightarrow^*$.

A *computation* is a finite sequence of transitions between configurations of a system Π starting from $(L_1, \ldots, L_m)$.

In case Π is an accepting device ($\omega = in$), then the input is given by the number of conformons (counted with their multiplicity) present in L_z. The input is accepted by Π if it reaches a configuration in which any conformon is present in ack, halting in this way the computation.

Formally:

$$N(\Pi) = \{|L_z| \mid (L_1, \ldots, L_m) \Rightarrow^* (M'_1, \ldots, M'_m) \Rightarrow (M_1, \ldots, M_m), \\ supp(M'_{ack}) = \emptyset, supp(M_{ack}) \neq \emptyset\}.$$

In case Π is a generating device ($\omega = out$), then $supp(L_z) = \emptyset$. The result of a computation is given by M_z when any conformon is present in ack. When this happens the computation is halted and the number of conformons (counted with their multiplicity) present in M_z defines the *number generated* by Π.

Formally:

$$N(\Pi) = \{|M_z| \mid (L_1, \ldots, L_m) \Rightarrow^* (M'_1, \ldots, M'_m) \Rightarrow (M_1, \ldots, M_m), \\ supp(M'_{ack}) = \emptyset, supp(M_{ack}) \neq \emptyset\}.$$

In the conformon-P systems area, in general one uses graphical representations instead of formal definitions in order to specify systems appearing in examples or proofs. We recall now some conventions used in these representations – details can be found in the papers mentioned in the end of Introduction.

Membranes/compartments are represented by labelled ovals, having inside the associated conformons and interaction rules. Conformons present in the initial

configuration of a system are written in **bold** inside a membrane while the ones written in normal font are present in that compartment in one of the possible configurations of the system. A slash (/) between values in a conformon indicates that a conformon can have any of the indicated values. The multiplicity is indicated only for conformons which appear in more than one copy. Directed edges between compartments are represented as arrows with their predicate indicated close to them. Several edges connecting two compartments are depicted as just one edge with different predicates separated by a slash (/). For instance, Figure 1 presents a conformon-P system which accepts any positive even number (the input membrane is the one with label 1 and the acknowledging one is membrane 11).

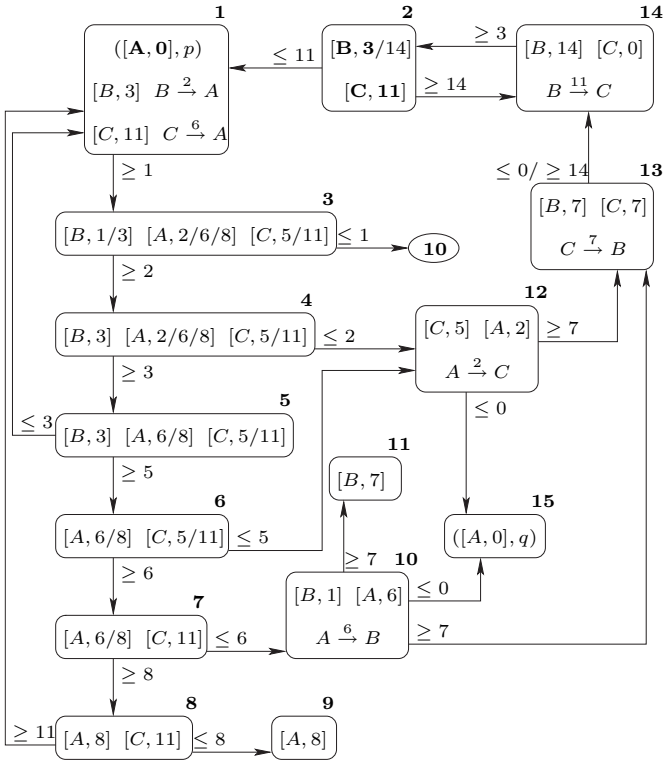

Fig. 1. A conformon-P system accepting even numbers.

In proofs there appear large conformon-P systems, that is why it is useful to consider *modules* which are sort of shortcuts of graphical representations. Such modules are explained in detail in several papers, e.g., in [3].

The basic modules are the *splitter* (it selects conformons depending on their values; specifically, when conformons of type $[a, p_i]$, $1 \leq i \leq h$, are present in a

given compartment, they can pass to specific different compartments depending on values p_i) and the *separator* (it selects conformons depending on their name; specifically, when conformons of type $[a_i, p]$, $1 \leq i \leq h$, are present in a compartment, they can pass to specific different compartments depending on a_i).

In the pictorial representations of conformon-P systems the modules are indicated by tick ovals, linked by arrows marked with predicates, which are of the form $= n_i$ in the case of splitters and of the form $[a, p_i]$ in the case of separators; usual membranes and arrows marked with predicates can be interleaved with modules. For instance, in Figure 2 we give a version of the system represented in Figure 1 where a splitter is also involved.

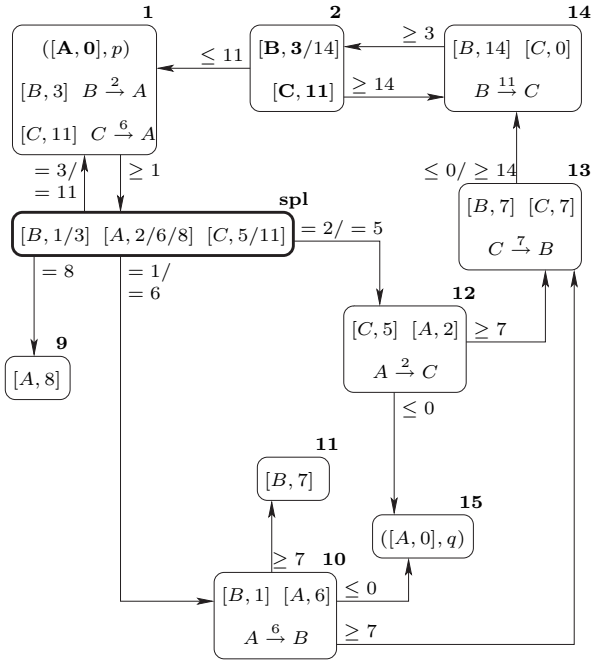

Fig. 2. The conformon-P system with a splitter associated to the system in Figure 1.

2.2 Asynchronous Tissue-like P Systems

We introduce the P systems of the form we have described in the Introduction, with a series of ingredients as presented before for conformon-P systems. Because we work only with asynchronous systems, from now on we omit mentioning this feature.

An *EC tissue-like P system* of degree m is a tuple

$$\Pi = (V, \mu, \omega_z, ack, L_1, \ldots, L_m, R_1, \ldots, R_m, P_1, \ldots, P_m),$$

where:

V is a finite alphabet whose elements are called *objects*;
$\mu = (Q, E)$ is a graph indicating the underlying compartment structure of Π, where
  $Q = \{1, \ldots, m\}$ is the set of *membranes/compartments*;
  $E \subseteq Q \times Q$ is the set of directed *edges* between compartments;
ω_z with $\omega \in \{in, out\}$ and $z \in Q$ indicates if Π is an accepting ($\omega = in$) or generating ($\omega = out$) device; the compartment z contains the input or output, respectively;
$ack \in Q$ indicates the *acknowledging* compartment;
$L_i : V \to \mathbb{N} \cup \{+\infty\}$, $1 \leq i \leq m$, are multisets of objects in V, with $supp(L_{ack}) = \emptyset$;
$R_i, 1 \leq i \leq m$, are sets of *evolution* rules of the form $ab \to cd$ with $a, b, c, d \in V$;
$P_i, 1 \leq i \leq m$, are sets of *communication* rules of the form (a, go) with $a \in V$.

A tissue-like P system is *cycle-free* if $ab \to cd \in R_i$ implies that $cd \to ab$ does not belong to R_i (with some abuse of notation we represent multisets by strings and all their permutations).

A *configuration* of Π is an m-tuple $(M_1, \ldots, M_m)$ of multisets over V. The m-tuple $(L_1, \ldots, L_m)$, is called *initial configuration* (in the initial configuration the acknowledge compartment does not contain any object) while any configuration having $supp(M_{ack}) \neq \emptyset$ is called *final configuration*. In a final configuration no operation is performed even if it could.

For two configurations $(M_1, \ldots, M_m), (M'_1, \ldots, M'_m)$ of Π we write $(M_1, \ldots, M_m) \Rightarrow (M'_1, \ldots, M'_m)$ indicating a *transition* from $(M_1, \ldots, M_m)$ to $(M'_1, \ldots, M'_m)$, that is, the application of one rule in a compartment according to the following. If $a, b \in M_i$ and $ab \to cd \in R_i$, then $M'_i = M_i - \{a, b\} \cup \{c, d\}$. If $a \in M_i$ and $(a, go) \in P_i$, then $M'_i = M_i - \{a\}$, $M'_j = M_j \cup \{a\}$ if $(i, j) \in E$. If no rule is applied to a multiset M_i, then $M'_i = M'_i$. The reflexive and transitive closure of $\Rightarrow$ is indicated by $\Rightarrow^*$. If in a configuration a symbol can be subject to more than one rule, then one of them is non-deterministically applied.

A *computation* is a finite sequence of transitions between configurations of the system Π starting from $(L_1, \ldots, L_m)$.

In case Π is an accepting device ($\omega = in$), then the input is given by the number of symbols (counted with their multiplicity) present in L_z. The input is accepted by Π if it reaches a configuration in which any conformon is present in ack, halting in this was the computation.

Formally,

$$N(\Pi) = \{|L_z| \mid (L_1, \ldots, L_m) \Rightarrow^* (M'_1, \ldots, M'_m) \Rightarrow (M_1, \ldots, M_m), \\ supp(M'_{ack}) = \emptyset, supp(M_{ack}) \neq \emptyset\}.$$

In case Π is a generating device ($\omega = out$), then $supp(L_z) = \emptyset$. The result of a computation is given by M_z when any symbol is present in ack. When this

happens the computation is halted and the number of symbols (counted with their multiplicity) present in M_z defines the *number generated* by Π.

Formally,

$$N(\Pi) = \{|M_z| \mid (L_1, \ldots, L_m) \Rightarrow^* (M'_1, \ldots, M'_m) \Rightarrow (M_1, \ldots, M_m), \\ supp(M'_{ack}) = \emptyset, supp(M_{ack}) \neq \emptyset\}.$$

As usual in P systems (i.e., without separating evolution from communication), we avoid rules of the form (a, go) and associate target indication directly to evolution rules: an object which has to be communicated will appear in the right hand side of a rule paired with *go* (the objects without such a pair remain in the same membrane). Note the important detail that this time the communication of an object c appearing in the form (c, go) in a rule must be done immediately, this does not mean application of a rule, but it is just part of using the evolution rule. This is a difference with respect to conformon-P systems and to EC tissue-like P systems, but in the proofs below we will not have to take care of this aspect: communication will be done by evolution rules of the form $a \to (a, go)$ which are directly associated with communication rules of the form (a, go).

3 Computing with Conformon-P Systems

We recall now some results concerning the computing power of conformon-P systems. Proofs can be found, e.g., in [3].

A conformon-P systems is called *value-restricted* (in short, VR) if in its initial configuration all conformons present in an unbounded number of copies have value 0. In this way, the total value of conformons present in the system at any step of a computation is finite.

Theorem 1. *The family of sets of numbers generated by VR conformon-P systems coincides with the family of sets of numbers generated by partially blind register machines.*

The conformon-P system which can simulate a partially blind register machine is based on the construction indicated in Figure 3. We recall it because later we will point out some basic features of this construction useful in inferring results about (asynchronous) tissue-like P systems.

From Theorem 2 in [5] we know that if in the conformon-P system described in the previous theorem either priorities, maximal concurrency, or maximal parallelism are added, then the resulting systems are computationally complete.

Theorem 2. *The family of sets of numbers generated by VR conformon-P systems where evolution has priority on communication (if a conformon can be subject of an interaction rule and it can also pass to another membrane, then the interaction should be done) coincides with the family of sets of numbers generated by register machines (hence with the family of Turing computable sets of numbers).*

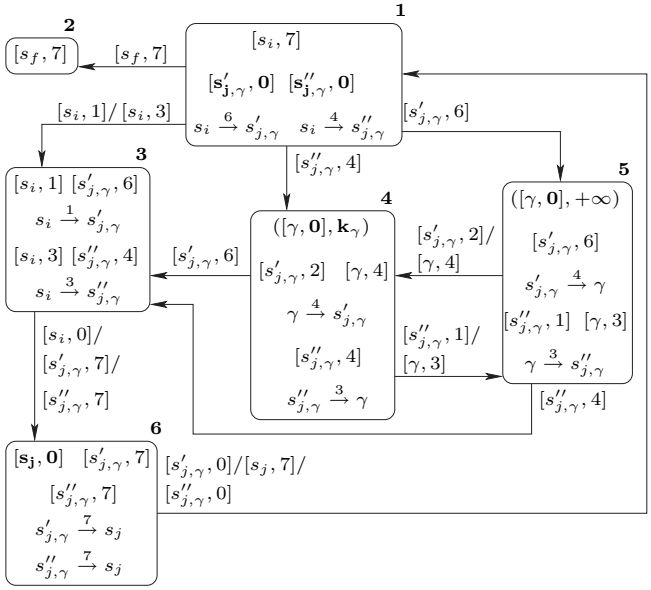

Fig. 3. The conformon-P system related to Theorem 1.

Also for this case we recall – in Figure 4 – the construction used in proving that a conformon-P system with priority as above can simulate a register machine.

4 From Conformon- to Tissue-like P Systems

First, let us point out a direct passage from conformon-P systems to EC tissue-like P systems.

Theorem 3. *Given any VR conformon-P system* $\Pi = (V, \mu, \omega_z, ack, L_1, \ldots, L_m, R_1, \ldots, R_m)$*, we can construct an EC tissue-like P system* $\Pi' = (V', \mu', \omega_z, ack, L'_1, \ldots, L'_m, R'_1, \ldots, R'_m, P'_1, \ldots, P'_m)$ *such that* $N(\Pi') = N(\Pi)$.

Proof. Consider a conformon-P system Π as above, with $\mu = (Q, E)$; denote by S the sum of the values of the conformons in Π. We construct the tissue-like P system Π' with:

$V' = \{a_p \mid a \in V, 0 \leq p \leq S\}$;
$\mu' = (Q, E')$ with $(i, j) \in E'$ for each $(i, j, pred) \in E$;
$L'_i(a_p) = k$ if $L_i([a, p]) = k$ for $1 \leq i \leq m$;
$a_p b_q \to a_{p-e} b_{q+e} \in R'_i$ if $a \xrightarrow{e} b \in R_i$, $0 \leq p, q \leq S, p \geq e$;
$(a_p, go) \in P'_i$ if $(i, j, \geq r) \in E$ for $r \leq p \leq S$ or $(i, j, \leq r) \in E$ for $0 \leq p \leq r$.

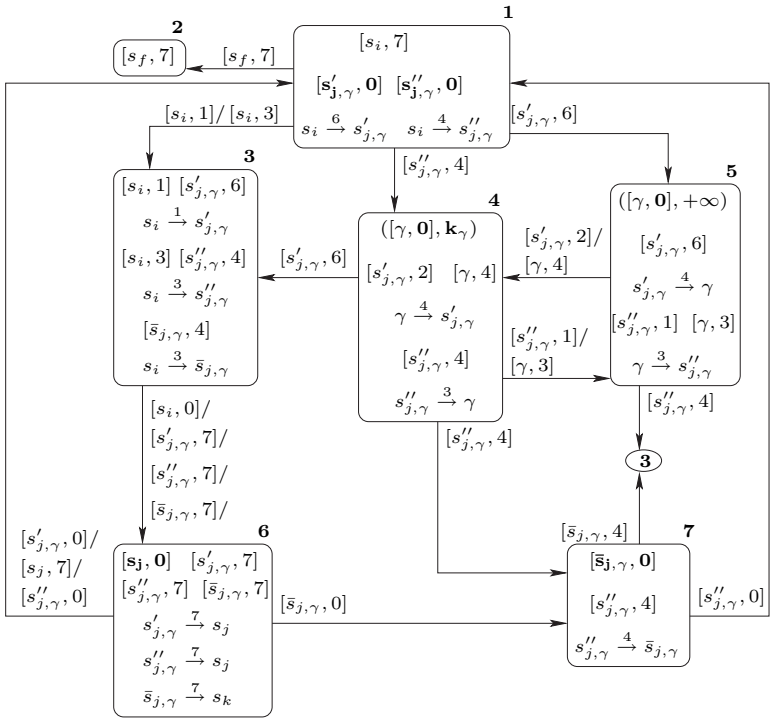

Fig. 4. The conformon-P system with priorities related to Theorem 2.

As V is by definition a finite alphabet and Π has finite total value, then the cardinality of V' is finite and equivalent to $S|V|$. Similarly, also the cardinality of the sets of rules is finite. It should be clear that the rules in the sets R'_i simulate the interaction between conformons, while the rules in the sets P'_i simulate the communication of conformons.

The initial configuration of Π is closely related to the one of Π': if n copies of the conformon $[a, p]$ are present in compartment i in Π, then n copies of the object a_p are present in compartment i of Π'. The system Π' simulates Π faithfully either if $\omega = in$ or $\omega = out$. An object will be present in the compartment ack of Π' if and only if a conformon can be present in the compartment ack of Π.

We conclude that $N(\Pi') = N(\Pi)$. □

The transcription-P system Π' constructed as in the proof of the previous theorem from a conformon-P system Π is cycle-free as soon as none of the compartments in Π contains rules of the form $a \xrightarrow{e} b$ and $b \xrightarrow{e} a$ for a and b conformons and $e \in \mathbb{N}$, and this is indeed the case for the systems constructed in the proofs of Theorems 1 and 2. Consequently, we have:

Corollary 1. *The family of sets of numbers generated by cycle-fee EC tissue-like P systems coincides with the family of sets of numbers generated by partially blind register machines.*

Like in the case of conformon-P systems, we can consider that also in tissue-like P systems the evolution rules have priority on communication rules. Then, from Theorem 2 we have:

Corollary 2. *The family of sets of numbers generated by cycle-free EC tissue-like P system with priorities coincides with the family of sets of numbers generated by register machines (hence with the family of Turing computable sets of numbers).*

The communication rules of tissue-like P systems can be avoided by adding targets to objects produced by evolution rules. In systems constructed as above, starting from conformon-P systems, we will get rules of the following forms: $ab \to (c, tar_1)(d, tar_2)$, with a, b, c, d symbols of the alphabet of Π and $tar_1, tar_2 \in \{here, in\}$, with the mentioning that *here* is omitted when specifying the rules.

Theorem 4. *Given any VR conformon-P system Π, we can construct a tissue-like P system Π' such that $N(\Pi') = N(\Pi)$.*

Proof. Given $\Pi = (V, \mu, \omega_z, ack, L_1, \ldots, L_m, R_1, \ldots, R_m)$, we construct $\Pi' = (V', \mu', \omega_z, ack, L'_1, \ldots, L'_m, U'_1, \ldots, U'_m)$ such that $N(\Pi') = N(\Pi)$ as follows. Let as assume that $\mu = (Q, E)$ and S is the sum of the values of the conformons in Π. Then:

$V' = \{a_p \mid a \in V, 0 \leq p \leq S\}$;
$\mu' = (Q, E')$ with $(i, j) \in E'$ for each $(i, j, pred) \in E$;
$L'_i(a_p) = k$ if $L_i([a, p]) = k$ for $1 \leq i \leq m$;
$a_p b_q \to a_{p-e} b_{q+e} \in U'_i$ if $a \xrightarrow{e} b \in R_i$, $0 \leq p, q \leq S, p \geq e$;
$a_p \to (a_p, go) \in U'_x$ if $(i, j, \geq r) \in E$ for $r \leq p \leq S$ or $(i, j \leq r) \in E$ for $0 \leq p \leq r$.

The system Π' simulates Π in a very similar way to the simulation described in the proof of Theorem 3. In the present proof, rules having *here* as both target indicators are equivalent to the interaction rules present in the proof of Theorem 3, and the remaining rules are equivalent to the communication rules present in the proof of Theorem 3.

Consequently, we obtain $N(\Pi') = N(\Pi)$. □

Similarly as before we have:

Corollary 3. *The family of sets of numbers generated by cycle-fee tissue-like P system coincides with the family of sets of numbers generated by partially blind register machines.*

If we assume that evolution rules having *here* as both target indicators have priority on the remaining rules, then we have:

Corollary 4. *The family of sets of numbers generated by cycle-fee tissue-like P system with priorities coincides with the family of sets of numbers generated by register machines (hence with the family of Turing computable sets of numbers).*

5 Concluding Remarks

Starting from some simple observations about the conformon-P systems which simulate (partially blind or arbitrary) register machines, we have inferred a series of results about the computing power of asynchronous tissue-like P systems, of the "standard" form (with targets associated with reaction products) and of the EC (evolution separated from communication) form. Cycle-free systems are obtained, which is an important feature for implementing P systems in a biochemical framework.

Some open problems remains to be considered. For instance, the tissue-like P systems deriving from the four corollaries we stated in the previous section have the same underlying graph (compartment structure) as the conformon-P systems depicted in Figure 3 and Figure 4, which, in turn, depend on the number of instructions of the register machines simulated by the respective conformon-P systems. Can the number of membranes be bounded (by a small number)? It is also worth trying to find interesting sets of numbers which can be computed (generated or accepted) by cycle-free systems as above.

References

1. M. Cavaliere: Evolution-communication P systems. In Gh. Păun, G. Rozenberg, A. Salomaa, C. Zandron, eds., *Membrane Computing, International Workshop, WMC-CdeA 2002, Curtea de Argeş, Romania, August 2002, Revised Papers*, LNCS 2597, Springer-Verlag, 2003, 134–145.
2. D.W. Corne, P. Frisco: Dynamics of HIV infection studied with cellular automata and conformon-P systems. *BioSystems*, 91, 3 (2008), 531–544.
3. P. Frisco: The conformon-P system: A molecular and cell biology-inspired computability model. *Theoretical Computer Science*, 312, 2-3 (2004), 295–319.
4. P. Frisco: Infinite hierarchies of conformon-P systems. In H.J. Hoogeboom, Gh. Păun, G. Rozenberg, and A. Salomaa, eds., *Membrane Computing, 7th International Workshop, WMC 2006, Leiden, The Netherlands, July 17-21, 2006, Revised Selected, and Invited Papers*, LNCS 4361, Springer, Berlin, 2006, 395–408.
5. P. Frisco: A hierarchy of computational processes. *Technical report HW-MACS-TR-0059*, Heriot-Watt University, 2008.
6. P. Frisco: Conformon-P systems with negative values. In G. Eleftherakis, P. Kefalas, Gh. Păun, G. Rozenberg, A. Salomaa, eds., *Membrane Computing, 8th International Workshop, WMC 2007, Thessaloniki, Greece, June 2007, Revised Selected and Invited Papers*, LNCS 4860, Springer, Berlin, 2007, 21–32.
7. P. Frisco, D.W. Corne: Modeling the dynamics of HIV infection with conformon-P systems and cellular automata. In G. Eleftherakis, P. Kefalas, Gh. Păun, G. Rozenberg, A. Salomaa, eds., *Membrane Computing, 8th International Workshop, WMC 2007, Thessaloniki, Greece, June 2007, Revised Selected and Invited Papers*, LNCS 4860, Springer, Berlin, 2007, 21–32.
8. P. Frisco, 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*, Workshop on Theory and Applications of P

Systems, TAPS, Timişoara (Romania), September 26-27, 2005, IEEE Computer Society, 2005, 427–430.
9. R. Gershoni, E. Keinan, Gh. Păun, R. Piran, T. Ratner, S. Shoshani: Research topics arising from the (planned) P systems implementation experiment in Technion. In this volume.
10. D.E. Green, S. Ji: The electromechanical model of mitochondrial structure and function. In J. Schultz, B. F. Cameron, eds., *Molecular Basis of Electron Transport*, Academic Press, New York, 1972, 1–44.
11. 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 (1985), 395–426.
12. S. Ji: Free energy and information contents of conformons in proteins and DNA. *BioSystems*, 54 (2000), 107–214.
13. S. Ji: The Bhopalator: an information/energy dual model of the living cell (II). *Fundamenta Informaticae*, 49, 1-3 (2002), 147–165.
14. G. Kemeny, I.M. Goklany: Polarons and conformons. *Journal of Theoretical Biology*, 40 (1973), 107–123.
15. G. Kemeny, I.M. Goklany: Quantum mechanical model for conformons. *Journal of Theoretical Biology*, 48 (1974), 23–38.
16. Gh. Păun, Y. Sakakibara, T. Yokomori: P systems on graphs of restricted forms. *Publ. Math. Debrecen*, 60 (2002), 635–660
17. M.V. Volkenstein: The conformon. *Journal of Theoretical Biology*, 34 (1972), 193–195.