Open access • Journal Article • DOI:10.1007/S11047-008-9102-Z

Efficient simulation of tissue-like P systems by transition cell-like P systems

Daniel Díaz-Pernil, Mario J. Pérez-Jiménez, Álvaro Romero-Jiménez

Institutions: University of Seville

Published on: 01 Dec 2009 - Natural Computing (Springer Netherlands)

Topics: Membrane computing, P system, Time complexity, Theory of computation and Decision problem

Efficient simulation of tissue-like P systems by transition cell-like P systems

Daniel Díaz-Pernil · Mario J. Pérez-Jiménez · Alvaro Romero-Jiménez

Abstract In the framework of P systems, it is known that the construction of exponential number of objects in polynomial time is not enough to efficiently solve **NP**-complete problems. Nonetheless, it could be sufficient to create an exponential number of membranes in polynomial time. Working with P systems whose membrane structure does not increase in size, it is known that it is not possible to solve computationally hard problems (unless $\mathbf{P} = \mathbf{NP}$), basically due to the impossibility of constructing exponential number of membranes, in polynomial time, using only evolution, communication and dissolution rules. In this paper we show how a family of recognizer tissue P systems with symport/antiport rules which solves a decision problem can be efficiently simulated by a family of basic recognizer P systems solving the same problem. This simulation allows us to transfer the result about the limitations in computational power, from the model of basic cell-like P systems to this kind of tissue-like P systems.

Keywords P systems · Tissue P systems · Recognizer P systems · Symport/antiport rules · Efficient simulation of cellular systems

1 Introduction

Membrane Computing starts from the assumption that the processes taking place within the compartmental structure of a living cell can be interpreted as computations (Păun 2000). The computational devices in Membrane Computing are called *P systems*. Roughly speaking, a P system consists of a membrane structure, in the compartments of which one

D. Díaz-Pernil · M. J. Pérez-Jiménez (✉) · Á. Romero-Jiménez
Department of Computer Science and Artificial Intelligence, Research Group on Natural Computing, University of Sevilla, Sevilla, Spain
e-mail: marper@us.es

D. Díaz-Pernil
e-mail: sbdani@us.es

Á. Romero-Jiménez
e-mail: alvaro@us.es

places multisets of objects which evolve according to given rules in a synchronous, non-deterministic, maximally parallel manner.[1]

In recent years, many different models of P systems have been proposed. The most studied variants are characterized by a *cell-like* membrane structure, where the communication happens between a membrane and the surrounding one. In this model we have a set of nested membranes, in such a way that the graph of the neighborhood relation is a tree.

We shall focus here on another type of P systems, the so-called (because of their membrane structure) *tissue P Systems* (Martín-Vide et al. 2003). Instead of considering a hierarchical arrangement, membranes are modelled as nodes of an undirected graph. This variant has two biological inspirations: intercellular communication and cooperation between neurons. The common mathematical model of these two mechanisms is a net of processors dealing with symbols and communicating these symbols along channels specified in advance. The communication between cells is based on symport/antiport rules.[2]Symport rules move a number of objects across a membrane together in the same direction, whereas antiport rules move objects across a membrane in opposite directions.

Since the seminal definition of tissue P systems several research lines have been developed and other variants have arisen. One of the most interesting variants of tissue P systems was presented in Păun et al. 2004 where the definition of tissue P systems is combined with the corresponding one of P systems with active membranes, yielding the model of *tissue P systems with cell division*.

This model has been studied in depth in Díaz-Pernil (2008), where the importance of the cell division rules in order for the model to be able to efficiently solve computationally hard problems is shown. We present here a (fundamental) part of the proof of this result, namely the possibility of simulating, in an efficient manner, a family of tissue P systems without division rules (e.g. using only communication rules) solving a decision problem, by a family of basic P system solving the same problem.

The paper is organised as follows. In Sect. 2 we recall some definitions related to tissue P systems (further information can be found in the literature, see (P system 2003)). Section 3 is devoted to simulations of cellular systems in general, and to simulations of recognizer tissue P systems with symport/antiport rules by basic recognizer P systems, in particular. Finally, in the last section some remarks and future work directions are presented.

2 Recognizer tissue P systems

Let us recall that a decision problem, X, is a pair (I_X, θ_X) such that I_X is a language over a finite alphabet (whose elements are called *instances*) and θ_X is a total boolean function over I_X. There exists a natural correspondence between languages and decision problems in the following way. Each language L, over an alphabet Σ, has a decision problem, X_L, associated with it as follows: $I_{X_L} = \Sigma^*$, and $\theta_{X_L} = \{(u,1) \mid u \in L\} \cup \{(u,0) \mid u \in \Sigma^* \backslash L\}$. Conversely, given a decision problem $X = (I_X, \theta_X)$, the language L_X over the alphabet of I_X associated with it is defined as: $L_X = \{u \in I_X \mid \theta_X(u) = 1\}$.

[1] An informal overview can be found in Păun and Pérez-Jiménez (2003) and further bibliography at P system http://ppage.psystems.eu/.

[2] This method of communication for P systems was introduced in Păun and Păun (2002).

In order to solve decision problems by membrane systems and having in mind the relationship between the solvability of such problems and the recognition of the languages associated with them, we consider tissue P systems as *languages recognizer* devices.

Definition 2.1 A tissue P system of degree $q \geq 1$ with symport/antiport rules is a tuple

$$\Pi = (\Gamma, \Sigma, \Omega, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{\text{in}}, i_{\text{out}})$$

where Γ is a finite alphabet (called working alphabet) whose elements are called objects; Σ is a finite alphabet (called input alphabet) strictly contained in Γ; $\Omega \subseteq \Gamma \backslash \Sigma$ is a finite alphabet, describing the set of objects present in the environment in arbitrarily many copies each; $\mathcal{M}_1, \ldots, \mathcal{M}_q$ are strings over $\Gamma \backslash \Sigma$, describing the *multisets of objects* placed in the q cells of the system; $\mathcal{R}$ is a set of communication rules of the form $r = (i, u_1/u_2, j)$, for $i, j \in \{0, 1, \ldots, q\}, i \neq j$ and $u_1, u_2 \in \Gamma^*$; $i_{\text{in}} \in \{1, \ldots, q\}$ is the input cell, and $i_{\text{out}} \in \{0, 1, \ldots, q\}$ is the output cell.

For a rule $(i, u_1/u_2, j)$ we say that the maximum of lengths of u_1 and u_2 is the *weight* of the rule. When applying such a rule, the objects of the multiset represented by u_1 are sent from region i to region j and simultaneously the objects of the multiset u_2 are sent from region j to region i. In this description, the number 0 represents the environment of the system.

The rules of a system as above are used in the non-deterministic maximally parallel manner as customary in membrane computing. In each step, all cells which can evolve must evolve (more precisely, in each step we apply an applicable set of rules which is maximal, no further rule can be added to it).

A configuration of Π is a tuple $C = (M_0, M_1, \ldots, M_q)$, where M_0 is a multiset of objects over $\Gamma \backslash \Omega$ (the objects in the environment which are in finitely many copies), and $M_1, \ldots, M_q$ are multisets of objects over Γ (the objects in each cell of the system). For two configurations C_1, C_2 of Π we write $C_1 \Rightarrow_\Pi C_2$, and we say that we have a *transition* from C_1 to C_2, if we can pass from C_1 to C_2 by applying the rules from $\mathcal{R}$.

The initial configuration of the system is $(\emptyset, \mathcal{M}_1, \ldots, \mathcal{M}_q)$. For each multiset m over the input alphabet, the initial configuration of the system associated with it is $(\emptyset, \mathcal{M}_1, \ldots, \mathcal{M}_{i_{\text{in}}} + m, \ldots, \mathcal{M}_q)$. Then, m is an *input multiset* of every computation $\mathcal{C} = \{C_i\}_{i<r}$ such that C_0 is the initial configuration of Π associated with m.

All computations start from an initial configuration and proceed as stated above; only halting computations give a result, which is encoded by the number of objects in the output cell i_{out} in the last configuration. From now on, we will consider that the output is collected in the environment (that is, $i_{\text{out}} = 0$, and thus we will omit i_{out} in the definition of tissue P systems). This way, if Π is a tissue P system and $\mathcal{C} = \{C_i\}_{i<r}$ is a halting computation of Π, with $C_i = (M_{i,0}, M_{i,1}, \ldots, M_{i,q})$, then the answer of the computation $\mathcal{C}$ is

$$\text{Output}(\mathcal{C}) = \Psi_{\Gamma \backslash \Omega}(M_{r-1,0})$$

where Ψ is the Parikh function.

In order to use these computational devices for solving decision problems, *recognizer tissue P systems* are introduced.

Definition 2.2 A tissue P system with symport/antiport rules is a recognizer system if the following holds:

(1) The working alphabet Γ has two distinguished objects `yes` and `no`, present in at least one copy in some initial multisets $\mathcal{M}_1, \ldots, \mathcal{M}_q$, but not present in Ω.

(2) All computations halt.
(3) If $\mathcal{C} = \{C_i\}_{i<r}$ is a computation of Π, then either the object yes or the object no (but not both) must have been released into the environment, and only in the last step of the computation.

Given a recognizer tissue P system with symport/antiport rules, and a computation $\mathcal{C} = \{C_i\}_{i<r}$ of Π, we define the result of $\mathcal{C}$ as follows:

$$\text{Output}(\mathcal{C}) = \begin{cases} \texttt{yes}, & \text{if } \Psi_{\{\texttt{yes},\texttt{no}\}}(M_{r-1,0}) = (1,0) \\ & \wedge\ \Psi_{\{\texttt{yes},\texttt{no}\}}(M_{k,0}) = (0,0) \quad \text{for } k = 0,\ldots,r-2 \\ \texttt{no}, & \text{if } \Psi_{\{\texttt{yes},\texttt{no}\}}(M_{r-1,0}) = (0,1) \\ & \wedge\ \Psi_{\{\texttt{yes},\texttt{no}\}}(M_{k,0}) = (0,0) \quad \text{for } k = 0,\ldots,r-2 \end{cases}$$

That is, $\mathcal{C}$ is an accepting computation (respectively, rejecting computation) if the object yes (respectively, no) appears in the environment (only) in the halting configuration of $\mathcal{C}$.

3 Simulating tissue P systems by basic P systems

The goal of this section is to establish the limitations of recognizer tissue P systems with symport/antiport rules from the computational complexity point of view. Specifically, we will justify that this model of tissue P systems can be simulated (in a efficient manner) by basic recognizer P systems.

Let us recall that in P systems, evolution rules, communication rules and rules involving dissolution are called *basic rules*. That is, by applying this kind of rules the size of the structure of membranes does not increase. Hence, it is not possible to construct an exponential working space in polynomial time using only basic rules in a P system.

Definition 3.1. A recogniser P system is a P system such that: (a) the working alphabet contains two distinguished elements yes and no; (b) all computations halt; and (c) if $\mathcal{C}$ is a computation of the system, then either object yes or object no (but not both) must have been released into the output region of the system, and only in the last step of the computation.

When we try to simulate the behaviour of a recognizer tissue P system with symport/antiport rules by a recognizer basic P system we have the following difficulties:

- On one hand, tissue P systems have a set of individual cells connected between them through a virtual graph provided by the rules, every cell can communicate with the environment, and only symport/antiport rules will be considered. On the other hand, P systems work with a hierarchized set of membranes structured by a rooted tree, only the skin membrane can send objects to the environment, and only evolution rules will be considered.
  - To address this, the P system that performs the simulation will work with pairs encoding, simultaneously, the objects from the tissue P system and the cells where they are placed.
- On one hand, cells in tissue P systems always contain objects with finite multiplicity, but the environment contains arbitrarily many copies of some objects. On the other hand, membranes and the environment in P systems always have objects with finite multiplicity.

– To address this, in the P system that performs the simulation, the objects from the tissue P system present in the environment with infinite multiplicity will not be considered in an explicit manner, but will be used without problems whenever necessary.

Next, we precise the meaning of (efficient) simulations in the framework of cellular systems.

Definition 3.2 Let Π and Π' be recognizer cellular systems (cell-like and/or tissue-like). We say that Π' *efficiently simulates* Π if the following holds:

- Π' can be constructed from Π by a deterministic Turing machine working in polynomial time.
- There exists a bijective function, f, from the set $\mathrm{Comp}(\Pi)$ of computations of Π onto the set $\mathrm{Comp}(\Pi')$ of computations of Π' such that:
  - A computation $\mathcal{C} \in \mathrm{Comp}(\Pi)$ is an accepting computation if and only if $f(\mathcal{C}) \in \mathrm{Comp}(\Pi')$ is an accepting one.
  - There exists a polynomial $p(n)$ such that for each $\mathcal{C} \in \mathrm{Comp}(\Pi)$ we have $|f(\mathcal{C})| \leq p(|\mathcal{C}|)$.

Next, for every recognizer tissue P system with symport/antiport rules we design a basic recognizer P systems efficiently simulating it, according to Definition 3.2.

Definition 3.3 Let $\Pi = (\Gamma, \Sigma, \Omega, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{\mathrm{in}})$ be a recognizer tissue P system of degree $q \geq 1$ with symport/antiport rules. Let us consider the basic recognizer P system $S(\Pi) = (\Gamma', \Sigma', \mu, \mathcal{M}'_1, \mathcal{R}', i'_{\mathrm{in}})$ defined as follows:

- $\Gamma' = \{(a,i) : a \in \Gamma \wedge i \in \{1,\ldots,q\}\} \cup \{(a,0) : a \in \Gamma \backslash \Omega\} \cup \{\mathtt{yes}, \mathtt{no}\}$. The objects of $S(\Pi)$ are ordered pairs encoding objects of Π and cells where the objects are placed. From the environment, we only consider objects with finite multiplicity, that is, belonging to $\Gamma \backslash \Omega$.
- $\Sigma' = \{(a, i_{\mathrm{in}}) : a \in \Sigma\}$.
- $\mu = [\,]_1$.
- $\mathcal{M}'_1 = \sum_{i=1}^{q} \sum_{a \in \Gamma \backslash \Sigma} (a,i)^{\mathcal{M}_i(a)}$.

For each cell i of Π and for each object $a \in \Gamma \backslash \Sigma$ belonging to that cell, we consider in the membrane of $S(\Pi)$ the pair (a, i) with the same multiplicity.

- In the set $\mathcal{R}'$ the following rules associated with $S(\Pi)$ are included:
  - For each rule $r_{\Pi} \equiv (i, a_1 \ldots a_m / b_1 \ldots b_n, j) \in \mathcal{R}$ with $i, j \neq 0$, associated with Π, we consider the following rule (denoted by $r_{S(\Pi)}$) $(a_1,i)\ldots(a_m,i)(b_1,j)\ldots(b_n,j) \rightarrow (b_1,i)\ldots(b_n,i)(a_1,j)\ldots(a_m,j)$
  - For each rule $r_{\Pi} \equiv (i, a_1 \ldots a_m / b_1 \ldots b_n, 0) \in \mathcal{R}$ with $i \neq 0$, associated with Π, we consider the following rule (denoted by $r_{S(\Pi)}$) $(a_1,i)\ldots(a_m,i)(b_1,0)\ldots(b_s,0) \rightarrow (b_1,i)\ldots(b_n,i)(a_1,0)\ldots(a_r,0)$ where $a_1,\ldots,a_r,b_1,\ldots,b_s \notin \Omega$ and $a_{r+1},\ldots,a_m, b_{s+1},\ldots,b_n \in \Omega$.
  - For each rule $r_{\Pi} \equiv (0, a_1 \ldots a_m / b_1 \ldots b_n, i) \in \mathcal{R}$ with $i \neq 0$, associated with Π, we consider the following rule (denoted by $r_{S(\Pi)}$) $(a_1,0)\ldots(a_r,0)(b_1,i)\ldots(b_n,i) \rightarrow (b_1,0)\ldots(b_s,0)(a_1,i)\ldots(a_m,i)$ where $a_1,\ldots,a_r,b_1,\ldots,b_s \notin \Omega$ and $a_{r+1},\ldots,a_m, b_{s+1},\ldots,b_n \in \Omega$.

– $(\texttt{yes}, 0) \rightarrow (\texttt{yes}, \text{out}); (\texttt{no}, 0) \rightarrow (\texttt{no}, \text{out})$. These rules translate the answer provided by the system Π to an answer for the system $S(\Pi)$.

- $i'_{\text{in}} = 1$,that is, the membrane of the system is the input membrane.

Proposition 3.1 *Let Π be a recognizer tissue P system with symport/antiport rules. The system $S(\Pi)$ defined in 3.3 can be constructed from Π by a deterministic Turing machine working in polynomial time.*

Proof It suffices to note that the amount of resources needed to construct $S(\Pi)$ from Π is polynomial in the size of the initial resources of Π. Indeed,

- Sizes of the alphabets: $|\Gamma_{S(\Pi)}| = |\Gamma| \cdot q + |\Gamma \backslash \Omega| + 2$ *and* $|\Sigma_{S(\Pi)}| = |\Sigma|$.
- Initial number of membranes: 1.
- Initial number of objects in $S(\Pi)$: $|\mathcal{M}'_1| = \sum_{i=1}^{q} \sum_{a \in \Gamma \backslash \Sigma} \mathcal{M}_i(a)$.
- Total number of rules: $|\mathcal{R}'| = |\mathcal{R}| + 2$.
- The maximum weight of rules in $R_{S(\Pi)}$ is less than or equal to the maximum weight of the rules in R_Π. □

In what follows, Π represents a recognizer tissue P system with symport/antiport rules and $S(\Pi)$ the basic recognizer P system associated with it, according to Definition 3.3.

Definition 3.4 Let $C = (M_0, M_1, \ldots, M_q)$ be a configuration of Π. Then we define the configuration $S(C) = (M'_1, M'_{\text{env}})$ of $S(\Pi)$ as follows:

- $M'_1 = \sum_{i=1}^{q} \sum_{a \in \Gamma} (a, i)^{M_i(a)} + \sum_{a \in \Gamma \backslash \Omega} (a, 0)^{M_0(a)}$.
- $M'_{\text{env}} = \emptyset$.

Let us see that S can be considered a one-to-one correspondence between the configurations of Π and the configurations of $S(\Pi)$ where the environment is empty.

Let us note by $Conf(\Pi)$ (respectively, $Conf(S(\Pi))$) the set of all configurations of Π (respectively, $S(\Pi)$). We also denote by $Conf^{\neg env}(S(\Pi))$ the set of all configurations of $S(\Pi)$ where the environment is empty.

Proposition 3.2 *The function S from $Conf(\Pi)$ to $Conf^{\neg env}(S(\Pi)))$ assigning $S(C)$, to each configuration $C \in Conf(\Pi)$ according to Definition 3.4, is bijective.*

Proof Let $C_1 = (\mathcal{M}_{1,0}, \mathcal{M}_{1,1}, \ldots, \mathcal{M}_{1,q})$ and $C_2 = (\mathcal{M}_{2,0}, \mathcal{M}_{2,1}, \ldots, \mathcal{M}_{2,q})$ be configurations of Π such that $S(C_1) = (\mathcal{M}'_1, \emptyset)$ and $S(C_2) = (\mathcal{M}'_2, \emptyset)$ are equal. Let us see that $C_1 = C_2$

- For each $a \in \Gamma$ we have $M_{1,i}(a) = M_{2,i}(a)$ $(1 \leq i \leq q)$.
- For each $a \in \Gamma \backslash \Omega$ we have $M_{1,0}(a) = M_{2,0}(a)$. By definition, for each $a \in \Omega$ we have $M_{1,0}(a) = M_{2,0}(a) = \infty$. So, $M_{1,0} = M_{2,0}$.

Let $C' = (M'_1, \emptyset)$ be a configuration of $S(\Pi)$ without objects in the environment. Let us consider the configuration $C = (M_0, M_1, \ldots, M_q)$ of Π defined as follows:

- $M_i(a) = M'_1(a, i)$, for each $a \in \Gamma$, $1 \leq i \leq q$.
- $M_0(a) = \begin{cases} M'_1(a, 0), & \text{if } a \in \Gamma \backslash \Omega \\ \infty, & \text{if } a \in \Omega \end{cases}$

From Definition 3.4 we deduce that $S(C) = C'$.

Next, we will see how this function can be extended to computations in both systems. For that, first we show that the function can be considered a *morphism* with respect to the relation $\Rightarrow_\Pi$.

Proposition 3.3 *Let C_1 and C_2 be configurations of Π. Then, $C_1 \Rightarrow_\Pi C_2$ if and only if $S(C_1) \Rightarrow_{S(\Pi)} S(C_2)$.*

Proof Let $C_1 = (\mathcal{M}_{1,0}, \mathcal{M}_{1,1}, \ldots, \mathcal{M}_{1,q})$ and $C_2 = (\mathcal{M}_{2,0}, \mathcal{M}_{2,1}, \ldots, \mathcal{M}_{2,q})$ be configurations of Π. We have,

$$S(C_1) = (M_1', \emptyset), \quad \text{with } M_1' = \sum_{i=1}^{q} \sum_{a \in \Gamma} (a,i)^{M_{1,i}(a)} + \sum_{a \in \Gamma \setminus \Omega} (a,0)^{M_{1,0}(a)}$$

$$S(C_2) = (M_2', \emptyset), \quad \text{with } M_2' = \sum_{i=1}^{q} \sum_{a \in \Gamma} (a,i)^{M_{2,i}(a)} + \sum_{a \in \Gamma \setminus \Omega} (a,0)^{M_{2,0}(a)}$$

Let us suppose that $C_1 \Rightarrow_\Pi C_2$. Then, there exists a multiset of rules m applicable to C_1 such that yields C_2. Let $S(m)$ be the following multiset of rules associated with the membrane of $S(\Pi)$:

$$S(m) = \sum_{r_\Pi \in \mathcal{R}} r_{S(\Pi)}^{m(r_\Pi)}$$

That is, for each rule r_Π of Π we add the rule $r_{S(\Pi)}$ in $S(m)$, with the same multiplicity. Then, it is easy to prove that $S(m)$ is a multiset of rules applicable to $S(C_1)$ transforming this configuration in $S(C_2)$. Now, let us suppose that $S(C_1) \Rightarrow_{S(\Pi)} S(C_2)$. Let $S(m)$ be an applicable multiset of rules producing $S(C_2)$ from $S(C_1)$. Bearing in mind that the environment of $S(C_2)$ is empty, we deduce that rules $(\mathtt{yes}, 0) \rightarrow (\mathtt{yes}, \mathrm{out})$, $(\mathtt{no}, 0) \rightarrow (\mathtt{no}, \mathrm{out})$ do not belong to $S(m)$. So, the rules in $S(m)$ are of type $r_{S(\Pi)}$ for some rule r_Π of Π. Let m be the following multiset of rules:

$$m = \sum_{r_{S(\Pi)} \in \mathcal{R}'} r_\Pi^{S(m)(r_{S(\Pi)})}$$

Then, it is easy to prove that m is an applicable multiset of rules producing C_2 from C_1. □

Every computation $\mathcal{C} = \{C_i\}_{i<r}$ of the recognizer tissue P system with symport/antiport rules Π is a halting computation, and either the object `yes` or the object `no` (buth no both) must have been released into the environment, and only in the last step. This information is encoded by the sequence of configurations $\{S(C_i)\}_{i<r}$ of $S(\Pi)$ by the occurrence either of the object $(\mathtt{yes}, 0)$ or the object $(\mathtt{no}, 0)$ in its membrane, and only in the configuration $S(C_{r-1})$. Hence, in order to obtain a halting computation in $S(\Pi)$ it is necessary to add an step that releases into the environment of $S(\Pi)$ either the object `yes` or the object `no`, applying either the rule $(\mathtt{yes}, 0) \rightarrow (\mathtt{yes}, \mathrm{out})$ or the rule $(\mathtt{no}, 0) \rightarrow (\mathtt{no}, \mathrm{out})$.

Definition 3.5 Let $\mathcal{C}$ be a computation of Π and let C be the halting configuration of $\mathcal{C}$. If $S(C) = (M_1', \emptyset)$, then we define the configuration $\mathrm{Last}(\mathcal{C})$ as follows:

$$\mathrm{Last}(\mathcal{C}) = \begin{cases} (M_1' - (\mathtt{yes}, 0), \mathtt{yes}), & \text{if } \mathcal{C} \text{ is an accepting computation} \\ (M_1' - (\mathtt{no}, 0), \mathtt{no}), & \text{if } \mathcal{C} \text{ is a rejecting computation} \end{cases}$$

Proposition 3.4 *Let $\mathcal{C} = \{C_i\}_{i<r}$ be a computation of Π. Then, the sequence of configurations $\mathcal{S}(\mathcal{C}) = \{C'_i\}_{i<r+1}$ given by*

$$C'_i = S(C_i), \quad \text{for } i<r; \quad C'_r = \text{Last}(\mathcal{C})$$

is a halting computation of $S(\Pi)$. Moreover, $|\mathcal{S}(\mathcal{C})| = |\mathcal{C}| + 1$, and $\mathcal{C}$ is an accepting computation if and only if $\mathcal{S}(\mathcal{C})$ is an accepting computation.

Proof From Proposition 3.3 we have $C'_i \Rightarrow_\Pi C'_{i+1}$, for $i < r-1$. Then, there is no rule of Π applicable to C_{r-1} and, either the object `yes` or the object `no` (but not both), have been released into the environment of Π at time $r-1$. So, either the object $(\texttt{yes}, 0)$ or the object $(\texttt{no}, 0)$ is in the membrane of $S(\Pi)$ in the configuration $S(C_{r-1}) = C'_{r-1}$,and it is the first time that it appears in that membrane. So, only either the rule $(\texttt{yes}, 0) \to (\texttt{yes}, \text{out})$ or the rule $(\texttt{no}, 0) \to (\texttt{no}, \text{out})$ is applicable to C'_{r-1}. Hence,

- $C'_{r-1} = S(C_{r-1}) \Rightarrow_{S(\Pi)} \text{Last}(\mathcal{C}) = C'_r$.
- C'_r is a halting configuration.
- The answers of the computations $\mathcal{C}$ and $\mathcal{S}(\mathcal{C})$ are equal.

That is, $\mathcal{C}$ is a halting computation if and only if $S(\mathcal{C})$ is a halting computation. Finally, note that $|\mathcal{S}(\mathcal{C})| = |\mathcal{C}| + 1$. □

Proposition 3.5 *Let $Comp(\Pi)$ and $Comp(S(\Pi))$ be the sets of all computations of the systems Π and $S(\Pi)$, respectively. Then, the function $\mathcal{S}$ from $Comp(\Pi)$ to $Comp(S(\Pi))$ assigning $\mathcal{S}(\mathcal{C}) \in Comp(S(\Pi))$ to each computation $\mathcal{C} \in Comp(\Pi)$, is a computational isomorphism in the following sense: each $\mathcal{C} \in Comp_\Pi$ is an accepting computation of Π if and only if $\mathcal{S}(\mathcal{C})$ is an accepting computation of $Comp(S(\Pi))$.*

Proof From Proposition 3.4 we deduce that the function $\mathcal{S}$ is well defined. Let $\mathcal{C}_1 = \{C_i\}_{i<r}$ and $\mathcal{C}_2 = \{C'_j\}_{j<s}$ be computations of Π such that $\mathcal{S}(\mathcal{C}_1) = \mathcal{S}(\mathcal{C}_2)$. Then,

- $r = s$ (we have $|\mathcal{S}(\mathcal{C}_1)| = r + 1$ and $|\mathcal{S}(\mathcal{C}_2)| = s + 1$).
- $S(C_i) = S(C'_i), \quad$ for each $i<r$.

So, $C_i = C'_i$ for each $i < r$; that is, $\mathcal{C}_1 = \mathcal{C}_2$. Hence, the function $\mathcal{S}$ is injective.

Let $\mathcal{C}' = \{C'_i\}_i$ be a computation of $S(\Pi)$. Let us show that there is a computation $\mathcal{C}$ of Π such that $S(\mathcal{C}) = \mathcal{C}'$

First, let us notice that there exists a configuration C' of $\mathcal{C}'$ such that $C' \notin \text{Conf}^{\neg env}_{S(\Pi)}$. Indeed, otherwise $\{S^{-1}(C'_i)\}_{i<r}$ is a computation of Π, so either the object `yes` or the object `no` must have appeared in the environment of Π in the configuration $S^{-1}(C'_{r-1})$. Hence, either the rule $(\texttt{yes}, 0) \to (\texttt{yes}, \text{out})$ or the rule $(\texttt{no}, 0) \to (\texttt{no}, \text{out})$ could be applied to the configuration C'_{r-1} and then it would not be a halting configuration. Let $j_0 = min\{i<r : C'_i \notin \text{Conf}^{\neg env}_{S(\Pi)}\}$. Then, $j_0 > 0$, because the environment is empty in the initial configuration.

Let us consider the sequence of configurations $\mathcal{C} = \{C_i\}_{i<j_0}$ of Π, given by $C_i = S^{-1}(C'_i)$, for each $i < j_0$. From Proposition 3.3 we have $C_i \Rightarrow_\Pi C_{i+1}$, for each $i < j_0-1$. From the construction of $S(\Pi)$ we deduce that only in the case that either the rule $(\texttt{yes}, 0) \to (\texttt{yes}, out)$ or the rule $(\texttt{no}, 0) \to (\texttt{no}, \text{out})$ would have been applied to the configuration C'_{j_0-1}, the environment would be nonempty in the configuration C'_{j0}. But this case only is possible if in the configuration C_{j_0-1} of Π the environment would contain either the object `yes` or the object `no`. Hence,

- C_{j_0-1} is a halting configuration of Π, since Π is a recognizer P system.
- $\mathcal{C}$ is a computation of Π.
- $C'_{j_0} = \mathrm{Last}(\mathcal{C})$.
- $j_0 = r-1$.
- $S(\mathcal{C}) = \mathcal{C}'$.

Thus, the function $\mathcal{S}$ is suprajective. From Proposition 3.4 we deduce that each computation $\mathcal{C}$ of Π is an accepting computation if and only if $\mathcal{S}(\mathcal{C})$ is an accepting computation of $S(\Pi)$. □

Corollary 3.1 *Let Π be a recognizer tissue P system with symport/antiport rules. Then, the system $S(\Pi)$ given in Definition 3.3 is a basic recognizer P system.*

Proof By construction, $S(\Pi)$ is a basic P system whose working alphabet contains the distinguished objects `yes` and `no`.

Let $\mathcal{C}'$ be a computation of $S(\Pi)$. By Proposition 3.5 there exists a computation $\mathcal{C}$ of Π such that $\mathcal{C}' = \mathcal{S}(\mathcal{C})$. From Proposition 3.4 we have $\mathcal{C}'$ is a halting computation. Moreover, $\mathcal{C}'$ is an accepting computation if and only if $\mathcal{C}$ is an accepting one. □

Corollary 3.2 *For each multiset m over the input alphabet of Π we define the multiset $S(m) = \sum_{a\in\Sigma}(a, i_{\mathrm{in}})^{m(a)}$. Then, m is an input multiset of a computation $\mathcal{C}$ of Π if and only if $S(m)$ is an input multiset of the computation $S(\mathcal{C})$ of $\mathcal{S}(\Pi)$.*

Proof Let us suppose that $\mathcal{C} = \{C_i\}_{i<r}$ is a computation of Π and let $C_0 = (M_0, M_1, \ldots, M_q)$ be its initial configuration. If m is the input multiset of $\mathcal{C}$, then $M_{i_{\mathrm{in}}} = \mathcal{M}_{i_{\mathrm{in}}} + m$. By the definition of $\mathcal{S}(\mathcal{C}) = \{C'_i\}_{i<r+1}$ we deduce that $C'_0 = (M', \emptyset)$ with

$$M' = \sum_{a\in\Gamma\setminus\Sigma}(a, i_{\mathrm{in}})^{M_{i_{\mathrm{in}}}(a)} + \sum_{a\in\Sigma}(a, i_{\mathrm{in}})^{m(a)}$$

Hence, m is the input multiset of the computation $\mathcal{C}$ of Π if and only if $S(m)$ is the input multiset of the computation $S(\mathcal{C})$ of $S(\Pi)$.□

We have shown that any family of recognizer tissue P systems with symport/antiport rules can be efficiently simulated by a family of basic recognizer P systems. This leads to the conclusion that the classes of decision problems that can be efficiently solved (see Pérez-Jiménez et al. 2003 for a precise definition of this concept) by both models, are the same.

Let us recall that only tractable problems can be efficiently solved by basic recognizer P systems (Gutiérrez-Naranjo et al. 2006). Thus recognizer tissue P systems with symport/antiport rules cannot efficiently solve computationally hard problems (unless **P** = **NP**).

4 Final remarks and future work

In this paper, we have shown that one model of tissue-like P systems can be efficiently simulated by basic cell-like P systems. It would be interesting to extend the simulation to another kind of tissue P systems and to analyse the converse: simulation of P systems by tissue-like devices.

As a consequence, some limitations of recognizer tissue P systems with symport/antiport rules from the computational efficiency point of view can be derived. It is well known

that by adding division rules we can solve the SAT problem in linear time by using communication rules with weight at most two (Păun et al. 2004). In order to obtain new borderlines between tractability and intractability of problems we propose to study the possibility to restrict the use of communication rules to only weight one, or substituting division rules for rules capturing in tissue P systems the underlying ideas of membrane creation rules of P systems.

Further interesting topics would be assigning a new role to the environment of tissue P systems. Specifically considering that only objects with finite multiplicity can be included in the environment in any moment. It seems that this new scenario would be equivalent to tissue P systems without environment. Is it possible to solve **NP**-complete problems in polynomial time in this new framework, permitting division rules?

Acknowledgements The authors acknowledge the support of the project TIN2006-13425 of the Ministerio de Educación y Ciencia of Spain, cofinanced by FEDER funds, and the support of the project of excellence TIC-581 of the Junta de Andalucía.

References

Díaz-Pernil D (2008) Sistemas celulares de tejidos: Formalización y eficiencia computacional. PhD Thesis, University of Sevilla

Gutiérrez-Naranjo MA, Pérez-Jiménez MJ, Riscos-Núnez A, Romero-Campero FJ, Romero-Jiménez A (2006) Characterizing tractability by cell-like membrane systems. In: Subramanian KG, Rangarajan K, Mukund M (eds) Formal models, languages and applications. World scientific, series in machine perception and artificial intelligence, vol 66, chapter 9, pp 137–154

Martín-Vide C, Pazos J, Păun Gh, Rodríguez Patón A (2003) Tissue P systems. Theor Comput Sci 296:295–326

P systems http://ppage.psystems.eu/

Păun Gh (2000) Computing with membranes. J Comput Syst Sci 61(1):108–143

Păun Gh (2002) Membrane computing. An introduction. Springer-Verlag, Berlin

Păun A, Păun Gh (2002) The power of communication: P systems with symport/antiport. New Generat Comput 20(3):295–305

Păun Gh, Pérez-Jiménez MJ (2003) Recent computing models inspired from biology: DNA and membrane computing. Theoria 18(46):72–84

Păun Gh, Pérez-Jiménez MJ, Riscos-Núnez A (2004) Tissue P systems with cell division. In: Păun Gh, Riscos-Núnez A, Romero-Jiménez A, Sancho-Caparrini F (eds) Second brainstorming week on membrane computing, Sevilla, Report RGNC 01/2004, pp 380–386

Pérez-Jiménez MJ, Romero-Jiménez A, Sancho-Caparrini F (2003) Complexity classes in cellular computing with membranes. Nat Comput 2(3):265–285