A P–Lingua based Simulator for Tissue P Systems with Cell Separation

Ignacio PÉREZ–HURTADO, Luis VALENCIA–CABRERA, José M. CHACÓN, Agustín RISCOS–NÚÑEZ, Mario J. PÉREZ–JIMÉNEZ

Research Group on Natural Computing
Dpt. of Computer Science and Artificial Intelligence, University of Sevilla Avda. Reina Mercedes s/n. 41012 Sevilla, Spain

E-mail: {perezh,lvalencia}@us.es, jmchaconl@gmail.com, {ariscosn,marper}@us.es

Abstract. Tissue P systems are one of the currently active research topics within the field of Membrane Computing. In particular, their computational efficiency is being investigated in the case when the number of cells can grow by means of *cell separation* rules. In order to complement this study, it is useful to provide simulation software tools for this variant of tissue P systems.

The paper presents an extension of P-Lingua specification language, in order to include the class of *tissue P systems with cell separation*. This extension involves updating the P-Lingua parser, by reinterpreting some operators, along with some new ingredients. In addition, a new built-in simulation algorithm that has been added to the core library of P-Lingua is also presented.

A case study of a family of tissue P systems with cell separation of type **TSC**(**3**) solving SAT is used to show the dynamics of the simulator.

1. Introduction

Membrane Computing provides a framework for designing distributed parallel models inspired by some basic features of biological membranes. Since Păun introduced it in [6], many different classes of P systems have been investigated. Most of them are computationally complete/universal, that is, equivalent in power to Turing machines, as well as computationally efficient, i.e., are able to trade space for time

and solve in this way computationally hard problems in a feasible time. Some variants of P systems are known as tissue P systems, based on a representation of membranes placed in the nodes of a graph, inspired from the cell inter-communication in tissues. Since the initial definition of tissue P systems [4] several research lines have been developed and new variants have arisen. One of the most interesting variants of tissue P systems was presented in [7] where the definition of tissue P systems is combined with P systems with active membranes, yielding the model of tissue P systems with cell division. In [7] the first uniform and polynomial-time solution for the SAT problem through a family of tissue P systems is given.

Since tissue P systems are distributed parallel computing devices, it is necessary to design software applications in order to simulate such devices and to experimentally validate the tissue based models. In order to provide a general framework to specify, parse and simulate P systems, a programming language, P–Lingua [1, 11], was designed. The authors of P-Lingua developed a Java library providing several services, including parsers for input files and built-in simulators for cell-like P systems. This framework intends to cover as many variants of P systems as possible, so some additions in the language and a new simulator were included in P-Lingua [3] to permit the specification and simulation of tissue P systems with symport/antiport and cell division rules. In the present paper a new extension of P-Lingua is provided, including some syntactic additions to define tissue P systems with cell separation, along with a new built-in simulator to simulate computations of such new models.

The paper is structured as follows. In Section 2, we introduce some preliminary definitions about recognizer tissue P systems with symport/antiport rules and cell separation rules. Section 3 describes the extensions for the P-Lingua programming language in order to support tissue P systems. In Section 4, we introduce the simulator for tissue P systems used in this paper, including the simulation algorithm. Section 5 is devoted to a case study of simulation, based on the solution of SAT provided in [10]. Finally, conclusions and future work are discussed in Section 6.

2. Recognizer tissue P systems with cell separation

Tissue P systems with cell separation are inspired by the fact that alive tissues are not *static* networks of cells, since new cells are generated by cell fission in a natural way: a "parent" cell splits into two "child" cells, in such a way that the contents of the parent cell gets *distributed* between them.

First, we recall some preliminaries. An *alphabet* Γ is a non–empty set whose elements are called *symbols*. A *multiset* m over an alphabet Γ is a pair $m = (\Gamma, f)$ where $f : \Gamma \to \mathbb{N}$ is a mapping. If $m = (\Gamma, f)$ is a multiset then its *support* is defined as $supp(m) = \{x \in \Gamma \mid f(x) > 0\}$. A multiset is finite if its support is a finite set. If $supp(m) = \{a_1, \ldots, a_k\}$ then the multiset m will be denoted as $m = a_1^{f(a_1)} \ldots a_k^{f(a_k)}$ (here the order is irrelevant), and we say that $f(a_1) + \ldots + f(a_k)$ is the cardinal of m, denoted by $|m|$. The empty multiset is denoted by λ.

Let $m_1 = (\Gamma, f_1)$ and $m_2 = (\Gamma, f_2)$ multisets over Γ. The *union* of m_1 and m_2, denoted by $m_1 + m_2$ is the multiset (Γ, g), where $g = f_1 + f_2$, that is, $g(x) =$

$f_1(x) + f_2(x)$ for each $x \in \Gamma$. The *relative complement* of m_2 in m_1, denoted by $m_1 \setminus m_2$ is the multiset (Γ, g), where $g(x) = f_1(x) - f_2(x)$ if $f_1(x) \geq f_2(x)$ and $g(x) = 0$ otherwise.

Definition 1. A *tissue P system with cell separation* of degree $q \geq 1$ is a tuple

$$\Pi = (\Gamma, \Gamma_1, \Gamma_2, \Sigma, \mathcal{E}, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{in}, i_{out}),$$

where:

1. Γ is a finite *alphabet*.
2. $\{\Gamma_1, \Gamma_2\}$ is a partition of Γ, that is, $\Gamma = \Gamma_1 \cup \Gamma_2$, $\Gamma_1, \Gamma_2 \neq \emptyset$, $\Gamma_1 \cap \Gamma_2 = \emptyset$.
3. $\Sigma \subseteq \Gamma$ is the *input alphabet*.
4. $\mathcal{E} \subseteq \Gamma$ is an *environment alphabet*.
5. $\mathcal{M}_1, \ldots, \mathcal{M}_q$ are multisets over Γ.
6. $\mathcal{R}$ is a finite set of rules of the following forms:
   (a) *Communication rules*: $(i, u/v, j)$, where $i, j \in \{0, 1, \ldots, q\}, i \neq j$, $u, v \in \Gamma^*$, $|u| + |v| > 0$;
   (b) *Separation rules*: $[a]_i \rightarrow [\Gamma_1]_i[\Gamma_2]_i$, where $i \in \{1, \ldots, q\}$, $i \neq i_{out}$, $a \in \Gamma$.
7. $i_{in} \in \{1, \ldots, q\}$ and $i_{out} \in \{0, 1, \ldots, q\}$.

In other words, a *tissue P system with cell separation* of degree $q \geq 1$ can be viewed as a set of q cells, labeled by $1, \ldots, q$, with an environment labeled by 0 such that: (a) $\mathcal{M}_1, \ldots, \mathcal{M}_q$ are multisets over Γ representing the objects (elements in Γ) initially placed in the q cells of the system; (b) $\mathcal{E} \setminus \Sigma$ is the set of objects located initially in the environment of the system, all of them appearing in an *arbitrary number of copies*; and (c) i_{in} represents the input cell, and $i_{out} \in \{0, 1, \ldots, q\}$ represents the *region* where the output of the system will be sent (being either a distinguished cell when $i_{out} \in \{1, \ldots, q\}$, or the environment when $i_{out} = 0$).

When applying a rule $(i, u/v, j)$, the objects of the multiset u are sent from region i to region j and, simultaneously, the objects of multiset v are sent from region j to region i. A separation rule $[a]_i \rightarrow [\Gamma_1]_i[\Gamma_2]_i$ is applicable to cell i if object a is contained in that cell. When applying a separation rule $[a]_i \rightarrow [\Gamma_1]_i[\Gamma_2]_i$, in reaction with an object a, the cell i is separated into two cells with the same label; at the same time, object a is consumed; the objects from Γ_1 are placed in the first cell, those from Γ_2 are placed in the second cell; the output cell i_{out} cannot be separated.

The rules are used in a non-deterministic maximally parallel manner as customary in membrane computing. At each step, all cells which can evolve must evolve in a maximally parallel way, that is, we apply a multiset of rules which is maximal, no further applicable rule can be added, with the following important remark: when a

cell is separated, the separation rule is the only one which is applied for that cell at that step. The new cells resulting from separation could participate in the interaction with other regions by means of communication rules at the next step – provided that they are not separated once again. The label of a cell precisely identifies the rules which can be applied to it.

A *configuration* at any instant of Π is described by all multisets of objects over Γ associated with all the cells present in the system, and the multiset of objects over $\Gamma \setminus \mathcal{E}$ associated with the environment at that moment. Given a finite multiset m over Σ, the *initial configuration* with input m is $(\mathcal{M}_1, \ldots, \mathcal{M}_{i_{in}} + m, \ldots, \mathcal{M}_q; \emptyset)$. A configuration is a *halting configuration* if no rule of the system is applicable to it. We say that configuration $\mathcal{C}_1$ yields configuration $\mathcal{C}_2$ in one *transition step* if we can pass from $\mathcal{C}_1$ to $\mathcal{C}_2$ by applying the rules from $\mathcal{R}$ following the previous remarks. A *computation* of Π is a sequence of configurations such that: (a) the first term of the sequence is an initial configuration of the system (for a given input); (b) every other term of the sequence is obtained from the previous one by applying the rules of the system in a maximally parallel manner with the restrictions previously mentioned; and (c) either the sequence is infinite, or the last term of the sequence is a halting configuration (in this case we call it a *halting computation*).

A tissue P system is a *recognizer system* if all its computations verify the following property: either object `yes` or object `no` (but not both) must have been released into the environment, and only at the last step of the computation (hence, we also impose the condition that all computations halt for any possible input). We say that $\mathcal{C}$ is an *accepting* (respectively, *rejecting*) *computation* if object `yes` (respectively, object `no`) appears in the environment associated with the corresponding halting configuration of $\mathcal{C}$.

Let us recall that a *decision problem* is a pair (I_X, θ_X) where 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. Next, we define what means solving a decision problem in the framework of tissue P systems efficiently and in a uniform way. Bearing in mind that they provide devices with a finite description, a countable family of tissue P systems will be necessary in order to solve a decision problem.

Definition 2. We say that a decision problem $X = (I_X, \theta_X)$ is *solvable in a uniform way and polynomial time by a family* $\mathbf{\Pi} = \{\Pi(n) : n \in \mathbb{N}\}$ *of recognizer tissue P systems with cell separation* if the following holds:

1. The family $\mathbf{\Pi}$ is *polynomially uniform* by Turing machines, that is, there exists a deterministic Turing machine working in polynomial time which constructs the system $\Pi(n)$ from $n \in \mathbb{N}$.

2. There exists a pair (cod, s) of polynomial-time computable functions over I_X such that:

   (a) for each instance $u \in I_X$, $s(u)$ is a natural number and $cod(u)$ is an input multiset of the system $\Pi(s(u))$;

   (b) for each $n \in \mathbb{N}$, $s^{-1}(n)$ is a finite set;

(c) the family $\mathbf{\Pi}$ is *polynomially bounded* with regard to (X, cod, s), that is, there exists a polynomial function p, such that for each $u \in I_X$ every computation of $\Pi(s(u))$ with input $cod(u)$ is halting and it performs at most $p(|u|)$ steps;

(d) the family $\mathbf{\Pi}$ is *sound* with regard to (X, cod, s), that is, for each $u \in I_X$, if there exists an accepting computation of $\Pi(s(u))$ with input $cod(u)$, then $\theta_X(u) = 1$;

(e) the family $\mathbf{\Pi}$ is *complete* with regard to (X, cod, s), that is, for each $u \in I_X$, if $\theta_X(u) = 1$, then every computation of $\Pi(s(u))$ with input $cod(u)$ is an accepting one.

From the soundness and completeness conditions above we deduce that every P system $\Pi(n)$ is *confluent*, in the following sense: every computation of a system with the *same* input multiset must always give the *same* answer.

Let $\mathbf{R}$ be a class of recognizer tissue P systems. We denote by $\mathbf{PMC_R}$ the set of all decision problems which can be solved in a uniform way and polynomial time by means of families of systems from $\mathbf{R}$. The class $\mathbf{PMC_R}$ is closed under complement and polynomial–time reductions [9].

3. P-Lingua syntax for tissue P systems with cell separation

In the version of P-Lingua presented in [3], only one type of tissue P systems was allowed to be defined in P-Lingua files: tissue P systems with cell division. This is the starting point for the current syntax, that has been extended as explained below, in order to support tissue P systems with cell separation rules.

Definition of the P system model

In order to define a tissue P system with cell separation rules, the first line of the P-Lingua file should be as follows: `@model<TSCS>`

Definition of the partition of the working alphabet

In order to define the first part of the partition of the working alphabet, Γ_1, the following sentence should be written: `@ms1 += OBJECTS;` where `OBJECTS` is a comma-separated list of objects.

Similarly, in order to define the second part of the partition, Γ_2, an analogous sentence should be written, but using `@ms2` instead.

Actually, we need at least one sentence for each part, but the definition of the alphabets can be done by using several instructions, since the symbol `+=` indicates the parser to update the value of the variable on its left-hand side by adding the objects on the right-hand side to the previous list associated with it (if any). It is also allowed to use a single sentence with the symbol `=` instead. In this case the definitions are not made dynamically and in a cummulative way, but statically (a single comma-separated list for each part of the partition).

Definition of cell separation rules

Cell separation rules are defined with the syntax: `[a]'h --> []'h[]'h;` where `a` is the object that triggers the separation rule and `h` is the label of the cell to be separated.

Finally, the syntax of communication rules and the remaining elements of the language are the same as the ones used for the definition of tissue P systems with cell division rules [3].

4. A software simulator for tissue P systems with cell separation

The pLinguaCore library has been expanded to include a built-in simulator for tissue P systems with cell separation rules. This simulation algorithm is deterministic, and it reproduces only one possible computation of the defined P system. From a theoretical point of view, this behaviour should not be a problem, since the simulator is intended to be used on recognizer P systems. Recall that in the case of recognizer P systems from a family that solves a problem according to Definition 2, all computations for a given input must generate the same output, due to the confluence property.

The current release of pLinguaCore library is 4.0, and it can be downloaded from [11].

4.1. Simulation algorithm

The simulation algorithm described below generates one possible computation for a tissue P system with cell separation rules.

I. Initialization

1. Let C_0 be the initial configuration with q cells denoted by $c_1, \ldots, c_q$.
2. Let c_0 be a virtual cell with label 0 representing the environment, where all initial objects have infinite multiplicity.
3. Let $R_{sel} = \{\}$ be a set of tuples containing the selected rules to be executed at each step of computation, the identifiers of involved cells and the number of times that each rule will be executed.
4. Let $C_t = C_0$ be the current configuration.

II. Selection of communication rules

1. For each *communication rule* $(i, u/v, j)$ do
   (a) For each cell $c_{k_1} \in C_t$ with label i do

i. Let N be the greatest number such that the multiset of c_{k_1} contains N copies of the multiset u.

ii. For each cell $c_{k_2} \in C_t$ with label j, while $N > 0$ do

- Let M be the greatest number $M \leq N$ such that the multiset of c_{k_2} contains M copies of the multiset v
- Remove M copies of u from the multiset of c_{k_1}
- Remove M copies of v from the multiset of c_{k_2}
- Add $\langle c_{k_1}, c_{k_2}, (i, u/v, j), M\rangle$ to R_{sel}
- Let $N = N - M$

III. Selection of separation rules

1. For each *separation rule* $[a]_i \to [\Gamma_1]_i[\Gamma_2]_i$ do

   (a) For each cell $c_k \in C_t$ with label i such c_k does not appear in R_{sel} do

      i. If a is contained in the multiset of c_k, then

      - Remove one instance of a from the multiset of c_k
      - Add $\langle c_k, [a]_i \to [\Gamma_1]_i[\Gamma_2]_i\rangle$ to R_{sel}

IV. Execution of rules

1. For each tuple $\langle c_{k_1}, c_{k_2}, (i, u/v, j), M\rangle$ from R_{sel} do

   (a) Add M copies of v to the multiset of c_{k_1}

   (b) Add M copies of u to the multiset of c_{k_2}

2. For each tuple $\langle c_k, [a]_i \to [\Gamma_1]_i[\Gamma_2]_i\rangle$ from R_{sel} do

   (a) Create a new cell c'_k with label i and empty multiset.

   (b) Copy in the multiset of c'_k the objects of Γ_2 contained in the multiset of c_k

   (c) Remove from the multiset of c_k the objects of Γ_2

V. Ending

1. If $R_{sel} \neq \emptyset$, then

   - Let $C_{t+1} = C_t$
   - Let $R_{sel} = \{\}$
   - Go to **II**

2. **End**.

We assume that the set of rules is ordered. Moreover, for each step of the computation, the algorithm calculates a maximal multiset of communication rules applicable to the current configuration, and after that it goes through separation rules and selects them to be executed in that step, when possible, over cells not involved in any of the already selected communication rules. This strategy is motivated by the intuition that separation rules add more descriptive complexity than communication rules, so we decided to give them in a sense a "lower priority".

5. A case study: a family of tissue P systems with cell separation solving SAT

5.1. Description of the family

In this section, a solution of SAT problem running in polynomial time is presented, according to Definition 2, in the framework of tissue P systems with cell separation. To do so, a brute force algorithm has been designed. Starting from a certain formula given as an input, the algorithm is structured as follows:

- *Generation stage*: All possible truth assignments associated with the input formula are produced by using cell separation in an adequate way. Thus, an exponential amount of space (in terms of number of cells of the system) will be generated. Each cell codifies one truth assignment.
- *Checking stage*: Inside each cell, it is checked wether or not the formula is satisfiable by the truth assignment encoded by that cell.
- *Output stage*: The system sends the right answer to the environment, depending on the existence or not of some cell codifying a truth assignment making the given formula true.

In order to design a family of tissue P systems with cell separation solving SAT, a function is considered, codifying the pair of parameters characterizing the instances of the problem. These instances are the propositional formulas in simplified conjunctive normal form, and the needed parameters are the numbers of variables and clauses. So let us consider this polynomial-time computable function *pair function*: $\langle m, n\rangle = ((m+n)(m+n+1)/2)+m$, which is a primitive recursive and bijective function from $\mathbb{N}\times\mathbb{N}$ to $\mathbb{N}$. This way, for each natural number t there exist only two natural numbers m and n such that $t = \langle m, n\rangle$; that is, each natural number t identifies a number of variables and a number of clauses.

Next, the family $\mathbf{\Pi} = \{\Pi(t) : t \in \mathbb{N}\}$ of recognizer tissue P systems with cell separation from $\mathbf{TSC(3)}$ is defined, such that its rules have a maximum length of 3. Each system $\Pi(t)$ of the family will process all instances φ of SAT (that is, propositional formulas in simplified CNF form) with n variables and m clauses, where $t = \langle m, n\rangle$, provided that the appropriate input multiset $cod(\varphi)$ is supplied to the system. This multiset is placed in the corresponding input cell, and identifies the specific propositional formula with n variables and m clauses.

For each $(m,n) \in \mathbb{N} \times \mathbb{N}$, the following recognizer tissue P system with cell separation from $\mathbf{TSC}(\mathbf{3})$ is considered:

$$\Pi(\langle m,n \rangle) = (\Gamma, \Gamma_1, \Gamma_2, \Sigma, \mathcal{E}, \mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{R}, i_{in}, i_{out})$$

- The input alphabet is $\Sigma = \{x_{i,j}, \overline{x}_{i,j} : 1 \leq i \leq n\,,\, 1 \leq j \leq m\}$.
- The working alphabet is $\Gamma = \Gamma_1 \cup \Gamma_2$, where

$$\begin{array}{rcl}
\Gamma_1 & = & \{A_i, B_i : 1 \leq i \leq n+1\} \cup \{a_i, b_i, T_i, F_i, y_i, v_i, w_i : 1 \leq i \leq n\} \cup \\
& & \{c_i, t_i, f_i, s_i, z_i : 1 \leq i \leq n-1\} \cup \{E_j : 1 \leq j \leq m+1\} \cup \\
& & \{\alpha_i : 0 \leq i \leq 3n+2m+1\} \cup \{\beta_i : 0 \leq i \leq 3n+2m+2\} \cup \\
& & \{x_{i,j}, \overline{x}_{i,j}, e_{i,j}, \overline{e}_{i,j} : 1 \leq i \leq n\,,\, 1 \leq j \leq m\} \cup \\
& & \{d_{i,j,k}, \overline{d}_{i,j,k} : 1 \leq i \leq n\,,\, 1 \leq j \leq m\,,\, 1 \leq k \leq n\} \cup \\
& & \{q_{i,j}, r_{i,j}, u_{i,j} : 1 \leq i,j \leq n-1\} \cup \{q_0, S, \mathtt{yes}, \mathtt{no}\} \\
\\
\Gamma_2 & = & \{A'_i, B'_i : 1 \leq i \leq n+1\} \cup \{a'_i, b'_i, T'_i, F'_i : 1 \leq i \leq n\}
\end{array}$$

- The alphabet of the environment is:

$$\begin{array}{rcl}
\mathcal{E} & = & \{S\} \cup \{A_i, B_i, A'_i, B'_i : 2 \leq i \leq n+1\} \cup \\
& & \{T_i, F_i, F'_i, y_i, w_i : 1 \leq i \leq n\} \cup \{a_i, a'_i, b_i, b'_i, v_i : 2 \leq i \leq n\} \cup \\
& & \{T'_i, c_i, t_i, f_i, s_i, z_i : 1 \leq i \leq n-1\} \cup \{E_j : 1 \leq j \leq m+1\} \cup \\
& & \{\alpha_i : 1 \leq i \leq 3n+2m+1\} \cup \{\beta_i : 1 \leq i \leq 3n+2m+2\} \cup \\
& & \{q_{i,j}, r_{i,j}, u_{i,j} : 1 \leq i \leq n-1\,,\, 1 \leq j \leq n-1\} \cup \\
& & \{e_{i,j}, \overline{e}_{i,j} : 1 \leq i \leq n\,,\, 1 \leq j \leq m\} \cup \\
& & \{d_{i,j,k}, \overline{d}_{i,j,k} : 1 \leq i,k \leq n\,,\, 1 \leq j \leq m\}
\end{array}$$

- The initial multisets are: $\mathcal{M}_1 = A_1\, B_1$; $\mathcal{M}_2 = a_1\, a'_1\, b_1\, b'_1\, v_1\, q_{1,1}\, \alpha_0\, \mathtt{yes}\, \mathtt{no}$; and $\mathcal{M}_3 = \beta_0$.
- The set $\mathcal{R}$ consists of the following rules:

(1) $(1\,,\, A_i \,/\, a_i\, a'_i\,,\, 2)$, for $1 \leq i \leq n$, and $(1\,,\, A_{n+1} \,/\, E_1\,,\, 2)$.
(2) $(1\,,\, A'_i \,/\, a_i\, a'_i\,,\, 2)$, for $1 \leq i \leq n$, and $(1\,,\, A'_{n+1} \,/\, E_1\,,\, 2)$.

(3) $(1\,,\, B_i \,/\, b_i\, b'_i\,,\, 2)$, for $1 \leq i \leq n$.
(4) $(1\,,\, B'_i \,/\, b_i\, b'_i\,,\, 2)$, for $1 \leq i \leq n$.

(5) $(1\,,\, T_i \,/\, t_i\,,\, 2)$, for $1 \leq i \leq n-1$.
(6) $(1\,,\, T'_i \,/\, t_i\,,\, 2)$, for $1 \leq i \leq n-1$.

(7) $(1\,,\, F_i \,/\, f_i\,,\, 2)$, for $1 \leq i \leq n-1$.
(8) $(1\,,\, F'_i \,/\, f_i\,,\, 2)$, for $1 \leq i \leq n-1$.

(9) $(1\,,\, t_i \,/\, T_i\, T'_i,\, 0)$, for $1 \leq i \leq n-1$.
(10) $(1\,,\, f_i \,/\, F_i\, F'_i,\, 0)$, for $1 \leq i \leq n-1$.

(11) $(1\,,\, b_i \,/\, B_{i+1}\, S\,,\, 0)$, for $1 \leq i \leq n$, and $(1\,,\, B_{n+1} \,/\, \lambda\,,\, 0)$.
(12) $(1\,,\, b'_i \,/\, B'_{i+1}\,,\, 0)$, for $1 \leq i \leq n$, and $(1\,,\, B'_{n+1} \,/\, \lambda\,,\, 0)$.

(13) $(1\,,\,a_i \,/\, T_i\, A_{i+1}\,,\,0)$, for $1 \le i \le n$.
(14) $(1\,,\,a'_i \,/\, F'_i\, A'_{i+1}\,,\,0)$, for $1 \le i \le n$.

(15) $(2\,,\,A_i \,/\, c_i\,,\,0)$, for $1 \le i \le n-1$, and $(2\,,\,A_i \,/\, \lambda\,,\,0)$, for $n \le i \le n+1$.
(16) $(2\,,\,A'_i \,/\, c_i\,,\,0)$, for $1 \le i \le n-1$, and $(2\,,\,A'_i \,/\, \lambda\,,\,0)$, for $n \le i \le n+1$.

(17) $(2\,,\,B_i \,/\, c_i\,,\,0)$, for $1 \le i \le n-1$.
(18) $(2\,,\,B'_i \,/\, c_i\,,\,0)$, for $1 \le i \le n-1$.

(19) $(2\,,\,c_i \,/\, b_{i+1}\, b'_{i+1}\,,\,0)$, for $1 \le i \le n-1$.

(20) $(2\,,\,v_i \,/\, y_i^2\,,\,0)$, for $1 \le i \le n$.

(21) $(2\,,\,y_i \,/\, z_i\, w_i\,,\,0)$, for $1 \le i \le n-1$, and $(2\,,\,y_n \,/\, w_n\,,\,0)$.

(22) $(2\,,\,z_i \,/\, v_{i+1}\,,\,0)$, for $1 \le i \le n-1$.

(23) $(2\,,\,w_i \,/\, a_{i+1}\, a'_{i+1}\,,\,0)$, for $1 \le i \le n-1$, and $(2\,,\,w_n \,/\, E_1\,,\,0)$.

(24) $(2\,,\,q_{1,1} \,/\, r_{1,1}\,,\,0)$.

(25) $(2\,,\,q_{i,j} \,/\, r_{i,j}^2\,,\,0)$, for $1 \le i \le n-1,\ 2 \le j \le n-1$.

(26) $(2\,,\,r_{i,j} \,/\, s_i\, u_{i,j}\,,\,0)$, for $1 \le i, j \le n-1$.

(27) $(2\,,\,s_i \,/\, t_i\, f_i\,,\,0)$, for $1 \le i \le n-1$.

(28) $(2\,,\,u_{1,j} \,/\, q_{1,j+1}\, q_{2,j+1}\,,\,0)$, for $1 \le j \le n-2$.

(29) $(2\,,\,u_{i,j} \,/\, q_{i+1,j+1}\,,\,0)$, for $2 \le i, j \le n-2$.

(30) $(2\,,\,u_{i,n-1} \,/\, \lambda\,,\,0)$, for $1 \le i \le n-1$.

(31) $(2\,,\,T_i \,/\, \lambda\,,\,0)$, for $1 \le i \le n-1$.
(32) $(2\,,\,T'_i \,/\, \lambda\,,\,0)$, for $1 \le i \le n-1$.

(33) $(2\,,\,F_i \,/\, \lambda\,,\,0)$, for $1 \le i \le n-1$.
(34) $(2\,,\,F'_i \,/\, \lambda\,,\,0)$, for $1 \le i \le n-1$.

(35) $[\,S\,]_1 \longrightarrow [\,\Gamma_1\,]_1\ [\,\Gamma_2\,]_1$

(36) $(2\,,\,\alpha_i \,/\, \alpha_{i+1}\,,\,0)$, for $0 \le i \le 3n+2m$.

(37) $(3\,,\,\beta_i \,/\, \beta_{i+1}\,,\,0)$, for $0 \le i \le 3n+2m+1$.

(38) $(3\,,\,x_{i,j} \,/\, d_{i,j,1}^2\,,\,0)$, $(3\,,\,\bar{x}_{i,j} \,/\, \bar{d}_{i,j,1}^2\,,\,0)$, for $1 \le i \le n$, $1 \le j \le m$.

(39) $(3\,,\,d_{i,j,k} \,/\, d_{i,j,k+1}^2\,,\,0)$,
$(3\,,\,\bar{d}_{i,j,k} \,/\, \bar{d}_{i,j,k+1}^2\,,\,0)$, for $1 \le i \le n$, $1 \le j \le m$, $1 \le k \le n-1$.

(40) $(3\,,\,d_{i,j,n} \,/\, e_{i,j}, 0)$, $(3\,,\,\bar{d}_{i,j,n} \,/\, \bar{e}_{i,j}\,,\,0)$, for $1 \le i \le n$, $1 \le j \le m$.

(41) $(1\,,\, T_iE_j \,/\, e_{i,j}\,,\, 3)$, $(1\,,\, F_iE_j \,/\, \bar{e}_{i,j}\,,\, 3)$,
$(1\,,\, T'_iE_j \,/\, e_{i,j}\,,\, 3)$, $(1\,,\, F'_iE_j \,/\, \bar{e}_{i,j}\,,\, 3)$, for $1 \leq i \leq n$, $1 \leq j \leq m$.

(42) $(1\,,\, e_{i,j} \,/\, T_iE_{j+1}\,,\, 0)$,
$(1\,,\, \bar{e}_{i,j} \,/\, F_iE_{j+1}\,,\, 0)$, for $1 \leq i \leq n$, $1 \leq j \leq m-1$.

(43) $(1\,,\, e_{i,m} \,/\, E_{m+1}\,,\, 0)$, $(1\,,\, \bar{e}_{i,m} \,/\, E_{m+1}\,,\, 0)$, for $1 \leq i \leq n$.

(44) $(3,\, T_i \,/\, \lambda\,, 0)$, $(3\,,\, F_i \,/\, \lambda\,,\, 0)$,
$(3\,,\, T'_i \,/\, \lambda\,,\, 0)$, $(3\,,\, F'_i \,/\, \lambda\,,\, 0)$, for $1 \leq i \leq n$.

(45) $(3\,,\, E_j \,/\, \lambda\,,\, 0)$, for $1 \leq j \leq m$.

(46) $(1\,,\, E_{m+1} \,/\, \texttt{yes}\ \alpha_{3n+1+2m}\,,\, 2)$.

(47) $(1\,,\, \texttt{yes} \,/\, \beta_{3n+1+2m+1}\,,\, 3)$.

(48) $(2\,,\, \alpha_{3n+1+2m} \,/\, \beta_{3n+1+2m+1}\,,\, 3)$.

(49) $(2\,,\, \texttt{no}\ \beta_{3n+1+2m+1} \,/\, \lambda\,,\, 0)$.

(50) $(3\,,\, \texttt{yes} \,/\, \lambda\,,\, 0)$.

- The input cell is $i_{in} = 3$.
- The output region is the environment, $i_{out} = 0$.

5.2. A P-Lingua source code

The P–Lingua source code that defines a tissue P system belonging to the family specified above is available for download in the Examples section at [11].

5.3. Results

We have simulated several P systems of the defined family solving different instances of the SAT problem. In what follows, a table of selected results is provided, including the execution time for each simulation.

We have validated our simulator with 48 instances of the problem of different size and input. Grouping the formulas by the number of variables and clauses, we can calculate the average simulation times depending on the size of the instance of the problem. The result is shown in Table 2.

Table 1. Formulas – Satisfiability and simulation time

n	m	SAT	Time (s)	Formula
2	3	F	0.23	$(\bar{x}_1 + \bar{x}_2) \cdot x_1 \cdot x_2$
2	3	T	0.05	$(\bar{x}_1 + \bar{x}_2) \cdot x_2 \cdot (\bar{x}_1 + x_2)$
3	4	F	0.14	$(x_1 + x_2) \cdot (x_1 + x_2 + \bar{x}_3) \cdot \bar{x}_1 \cdot \bar{x}_2$
3	4	T	0.15	$(\bar{x}_1 + x_2) \cdot \bar{x}_1 \cdot x_3 \cdot (\bar{x}_1 + x_3)$
4	5	F	0.53	$(x_1 + x_4) \cdot (x_1 + \bar{x}_4) \cdot x_3 \cdot (x_2 + \bar{x}_3 + x_4) \cdot \bar{x}_1$
4	5	T	0.56	$(x_3 + \bar{x}_4) \cdot (\bar{x}_1 + x_2 + \bar{x}_3 + x_4) \cdot (x_1 + x_2) \cdot$ $(\bar{x}_1 + x_2 + x_3 + x_4) \cdot (\bar{x}_1 + x_3)$
5	6	F	2.16	$(x_1 + \bar{x}_2 + x_3 + x_5) \cdot (\bar{x}_1 + x_4) \cdot (\bar{x}_2 + \bar{x}_4) \cdot x_4 \cdot x_2 \cdot (\bar{x}_1 + x_2 + \bar{x}_3 + x_4)$
5	6	T	1.99	$(x_3 + x_4) \cdot (x_4 + \bar{x}_5) \cdot (\bar{x}_1 + x_2 + \bar{x}_3 + \bar{x}_4) \cdot (x_1 + \bar{x}_2 + x_4) \cdot$ $(x_1 + \bar{x}_3 + x_4) \cdot (x_3 + x_5)$
6	7	F	8.58	$(x_3 + x_5 + x_6) \cdot (x_3 + \bar{x}_4 + x_5 + \bar{x}_6) \cdot \bar{x}_3 \cdot \bar{x}_6 \cdot (x_1 + \bar{x}_2 + \bar{x}_3 + x_5 + x_6) \cdot (x_1 + x_4 + x_5) \cdot (\bar{x}_5 + x_6)$
6	7	T	8.07	$(\bar{x}_1 + \bar{x}_2 + x_5) \cdot (x_2 + x_3) \cdot (x_3 + \bar{x}_5 + \bar{x}_6) \cdot (\bar{x}_1 + x_2 + \bar{x}_3 + x_4 + x_5 + x_6) \cdot (\bar{x}_2 + \bar{x}_3) \cdot (x_2 + x_3 + x_6) \cdot (x_1 + \bar{x}_2 + x_3 + x_4 + x_5 + x_6)$
7	8	F	28.92	$(\bar{x}_5 + \bar{x}_6 + \bar{x}_7) \cdot (x_3 + \bar{x}_4 + x_7) \cdot (\bar{x}_1 + x_3 + x_5 + x_6 + \bar{x}_7) \cdot (x_1 + x_3 + \bar{x}_5 + x_6 + x_7) \cdot (x_2 + x_6) \cdot (x_2 + \bar{x}_6) \cdot \bar{x}_2 \cdot (x_2 + x_3 + x_4 + \bar{x}_5 + x_7)$
7	8	T	29	$(\bar{x}_2 + x_5 + x_6 + x_7) \cdot (x_2 + \bar{x}_4 + \bar{x}_5 + \bar{x}_7) \cdot (x_1 + x_2 + \bar{x}_3 + \bar{x}_6 + x_7) \cdot (x_1 + x_2 + x_3 + \bar{x}_5 + x_6 + \bar{x}_7) \cdot (\bar{x}_3 + \bar{x}_5 + x_6 + \bar{x}_7) \cdot (x_1 + x_2 + \bar{x}_3 + \bar{x}_7) \cdot (\bar{x}_1 + x_2 + \bar{x}_4 + \bar{x}_6) \cdot (x_3 + x_5 + x_6 + \bar{x}_7)$
8	9	F	119.3	$(x_3 + x_4 + \bar{x}_6 + \bar{x}_8) \cdot (x_6 + \bar{x}_7) \cdot (\bar{x}_2 + x_3 + \bar{x}_4 + x_5 + x_8) \cdot x_7 \cdot (x_1 + \bar{x}_2 + x_5 + \bar{x}_7 + \bar{x}_8) \cdot (x_2 + x_7 + x_8) \cdot (\bar{x}_6 + \bar{x}_7) \cdot (x_1 + x_5 + \bar{x}_8) \cdot (x_1 + \bar{x}_4 + x_5 + \bar{x}_6 + x_7)$
8	9	T	121.2	$(x_1 + \bar{x}_5 + \bar{x}_6 + \bar{x}_7 + \bar{x}_8) \cdot (x_2 + x_3 + x_4 + \bar{x}_6 + \bar{x}_7 + x_8) \cdot (x_3 + x_4 + \bar{x}_5 + \bar{x}_6 + \bar{x}_7 + \bar{x}_8) \cdot (\bar{x}_1 + \bar{x}_3 + \bar{x}_4 + \bar{x}_5 + x_6 + \bar{x}_7 + x_8) \cdot (\bar{x}_3 + \bar{x}_7) \cdot (x_4 + x_5 + \bar{x}_7) \cdot (x_1 + x_3 + \bar{x}_4) \cdot (x_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_4 + \bar{x}_5 + \bar{x}_6 + \bar{x}_7) \cdot (x_4 + \bar{x}_5 + \bar{x}_6 + x_7 + \bar{x}_8)$
9	10	F	406.95	$(\bar{x}_2 + \bar{x}_3 + x_5 + x_7) \cdot (x_2 + x_5 + x_6 + x_7 + x_9) \cdot (\bar{x}_3 + x_5 + x_7 + x_8) \cdot (x_1 + \bar{x}_4 + \bar{x}_5 + x_6 + x_8) \cdot (\bar{x}_2 + x_3 + x_5 + x_7 + x_8 + \bar{x}_9) \cdot (\bar{x}_2 + \bar{x}_4 + x_7 + x_9) \cdot (\bar{x}_2 + x_4 + x_6 + x_9) \cdot x_1 \cdot x_5 \cdot (\bar{x}_1 + \bar{x}_5)$
9	10	T	417.38	$(x_3 + x_8) \cdot (x_1 + \bar{x}_2 + x_5 + \bar{x}_6 + x_9) \cdot (x_3 + x_6 + x_9) \cdot (x_3 + x_5 + \bar{x}_6 + \bar{x}_8) \cdot (x_1 + x_2 + \bar{x}_5 + x_7 + \bar{x}_8 + \bar{x}_9) \cdot (\bar{x}_1 + x_2 + \bar{x}_4 + x_5 + \bar{x}_6 + \bar{x}_7 + x_9) \cdot (x_1 + x_2 + x_4 + \bar{x}_6 + x_8 + \bar{x}_9) \cdot (\bar{x}_1 + x_2 + \bar{x}_3 + \bar{x}_4 + x_7 + \bar{x}_8) \cdot (\bar{x}_1 + x_2 + x_3 + x_5 + \bar{x}_6 + x_8 + \bar{x}_9) \cdot (x_2 + \bar{x}_3 + x_4 + \bar{x}_6 + \bar{x}_7 + \bar{x}_9)$

Table 2. Simulation time depending on the size of the formula

n/m	2	3	4	5	6	7	8	9
3	0.5							
4		0.14						
5			0.52					
6				2.06				
7					8.57			
8						29.42		
9							117.64	
10								417.25

6. Conclusions and Future Work

In this paper a new extension of P-Lingua has been presented, adding a new variant, tissue P systems with cell separation. Besides, a new simulation algorithm has been designed and implemented, taking into account the special features of tissue P systems with symport/antiport rules and cell separation. This new simulator has been included into the library pLinguaCore, and checked by simulating a family of tissue P systems with cell separation taken from the literature: a uniform solution to SAT.

Since the new simulator is available, it could be interesting to develop or adapt visual environments oriented to users working with models based on tissue P systems with cell separation. These tools could provide an easy way to experimentally validate models which have been formally validated, such as the aforementioned solution to SAT problem. A possible approach could be the generation of a custom interface based on MeCoSim [8, 12].

Another promising pending work is related to the comparison of different models inside the framework of tissue P systems, such as tissue P systems with cell division and tissue P systems with cell separation. This comparison could lead to relevant conclusions regarding the practical tractability of **TSC**(**3**) and **TDC**(**3**) models, both from a theoretical point of view and regarding the running times of the simulators for the two variants. It is also interesting to investigate the effect of adding some alternative heuristic method of selecting the multiset of rules to be executed at each step.

On the other hand, it is important to note that the developed simulator is based on sequential technologies, so the inherent parallelism assumed by the theoretical model cannot be exploited. This limitation cannot be overcomed because a real implementation of P systems does not exist, but some performance improvement can be achieved by means of some parallel architectures and programming models, such as GPGPU. This technology has been successfully applied to some previous simulators, including one based on tissue P systems with cell division [2], so it could be interesting to develop a new one for cell separation.

Acknowledgements. The authors acknowledge the support of the project TIN2012–37434 of the Ministerio de Economía y Competitividad of Spain, and the support of the "Proyecto de Excelencia con Investigador de Reconocida Valía" of the Junta de Andalucía under grant P08–TIC04200, both cofinanced by FEDER funds.

References

[1] GARCÍA-QUISMONDO M., GUTIÉRREZ-ESCUDERO R., MARTÍNEZ-DEL-AMOR M.A., OREJUELA-PINEDO E., PÉREZ-HURTADO I., *P-Lingua 2.0: A software framework for cell-like P systems*, International Journal of Computers, Communications and Control, **4**(3), pp. 234–243, 2009.

[2] MARTÍNEZ-DEL-AMOR M.A., PÉREZ-CARRASCO J., PÉREZ-JIMÉNEZ M.J., *Simulating a Family of Tissue P Systems Solving SAT on the GPU*, Proceedings of the Eleventh Brainstorming Week on Membrane Computing, pp. 201–220, 2013.

[3] MARTÍNEZ-DEL-AMOR M.A., PÉREZ-HURTADO I., PÉREZ-JIMÉNEZ M.J., RISCOS-NÚÑEZ A., *A P-Lingua based simulator for tissue P systems*, Journal of Logic and Algebraic Programming, **79**(6), pp. 374–382, 2010 (doi: 10.1016/j.jlap.2010.03.009).

[4] MARTÍN-VIDE C., PAZOS J., PĂUN G., RODRÍGUEZ-PATÓN A., *Tissue P systems*, Theoretical Computer Science, **296**(2), pp. 295–326, 2003 (doi: 10.1016/S0304-3975(02)00659-X).

[5] PAN L., PÉREZ-JIMÉNEZ M.J., *Computational complexity of tissue–like P systems*, Journal of Complexity, **26**(3), pp. 296–315, 2010 (doi: 10.1016/j.jco.2010.03.001).

[6] PĂUN G., *Computing with membranes*, Journal of Computer and System Sciences, **61**(1), pp. 108–143, 2000 (doi: 10.1006/jcss.1999.1693). Was first circulated as Turku Center for Computer Science-TUCS Report No. 208 (1998).

[7] PĂUN G., PÉREZ-JIMÉNEZ M.J., RISCOS-NÚÑEZ A., *Tissue P systems with cell division*, International Journal of Computers Communications and Control, **3**(3), pp. 295–303, 2008.

[8] PÉREZ-HURTADO I., VALENCIA-CABRERA L., PÉREZ-JIMÉNEZ M.J., COLOMER M.A., RISCOS-NÚÑEZ A., *MeCoSim: A general purpose software tool for simulating biological phenomena by means of P systems*, IEEE 5th Int. Conf. on Bio-inpired Computing: Theories & Applications (BIC-TA 2010), pp. 637–643, 2010 (doi: 10.1109/BICTA.2010.5645199).

[9] PÉREZ-JIMÉNEZ M.J., ROMERO-JIMÉNEZ A., SANCHO-CAPARRINI F., *Complexity classes in models of cellular computing with membranes*, Natural Computing, **2**(3), pp. 265–285, 2003 (doi: 10.1023/A:1025449224520).

[10] Pérez-Jiménez M.J., Sosík P., *Improving the efficiency of tissue P systems with cell separation*, Proceedings of the Tenth Brainstorming Week on Membrane Computing, vol. **2**, pp. 105–140, 2012.

[11] The P–Lingua Web Site: `http://www.p-lingua.org/`

[12] MeCoSim Web Site: `http://www.p-lingua.org/mecosim`