INTERNATIONAL JOURNAL OF FOUNDATIONS OF COMPUTER SCIENCE

Volume 22, Number 1, January 2011

CONTENTS

Special Issue — Natural Computing: Theory and Applications

Preface 1
R. Freund, M. Gheorghe, S. Marcus, V. Mitrana and M. J. Perez-Jimmenez

On Strong Reversibility in P Systems and Related Problems 7
O. H. Ibarra

On String Languages Generated by Spiking Neural P Systems with Anti-Spikes 15
K. Krithivasan, V. P. Metta and D. Garg

Computation of Ramsey Numbers by P Systems with Active Membranes 29
L. Pan, D. Díaz-Pernil and M. J. Pérez-Jiménez

P Systems with Proteins on Membranes: A Survey 39
A. Păun, M. Păun, A. Rodríguez-Patón and M. Sidoroff

On a Partial Affirmative Answer for A Păun's Conjecture 55
I. Pérez-Hurtado, M. J. Pérez-Jiménez, A. Riscos-Núñez, M. A. Gutiérrez-Naranjo and M. Rius-Font

P Systems with Active Membranes Working in Polynomial Space 65
A. E. Porreca, A. Leporati, G. Mauri and C. Zandron

On the Power of Families of Recognizer Spiking Neural P Systems 75
P. Sosík, A. Rodríguez-Patón and L. Cienciala

Modeling Diffusion in a Signal Transduction Pathway: The use of Virtual Volumes in P Systems 89
D. Besozzi, P. Cazzaniga, S. Cocolo, G. Mauri and D. Pescini

Log-Gain Stoichiometric Stepwise Regression for MP Systems 97
V. Manca and L. Marchetti

A Simulation Algorithm for Multienvironment Probabilistic P Systems: A Formal Verification 107
M. A. Martínez-del-Amor, I. Pérez-Hurtado, M. J. Pérez-Jiménez, A. Riscos-Núñez and F. Sancho-Caparrini

An Overview on Operational Semantics in Membrane Computing 119
R. Barbuti, A. Maggiolo-Schettini, P. Milazzo and S. Tini

Formal Verification of P Systems Using Spin 133
F. Ipate, R. Lefticaru and C. Tudose

Small Universal TVDH and Test Tube Systems 143
A. Alhazov, M. Kogler, M. Margenstern, Y. Rogozhin and S. Verlan

Filter Position in Networks of Substitution Processors Does Not Matter 155
F. A. Montoro, J. Castellanos, V. Mitrana, E. Santos and J. M. Sempere

Functions Defined by Reaction Systems 167
A. Ehrenfeucht, M. Main and G. Rozenberg

P Systems and Topology: Some Suggestions for Research 179
P. Frisco and H. J. Hoogeboom

An Observer-Based De-Quantisation of Deutsch's Algorithm 191
C. S. Calude, M. Cavaliere and R. Mardare

PC Grammar Systems with Clusters of Components 203
E. Csuhaj-Varjú, M. Oswald and Gy. Vaszil

Orthogonal Shuffle on Trajectories 213
M. Daley, L. Kari, S. Seki and P. Sosìk

On the Number of Active Symbols in Lindenmayer Systems 223
J. Dassow and Gy. Vaszil

Positioned Agents in Eco-Grammar Systems 237
M. Langer and A. Kelemenová

Morphic Characterizations of Language Families in Terms of Insertion Systems and Star Languages 247
F. Okubo and T. Yokomori

Power Sums Associated with Certain Recursive Procedures on Words 261
A. Salomaa

Erratum 273

This journal is covered in SciSearch® (also known as Science Citation Index - Expanded), ISI Alerting Services, CompuMath Citation Index®, Current Contents/Engineering, Computing & Technology, MathSciNet, Mathematical Reviews, DBLP Bibliography Server and Zentralblatt MATH.

International Journal of Foundations of Computer Science
Vol. 22, No. 1 (2011) 107–118

DOI: 10.1142/S0129054111007873

World Scientific
www.worldscientific.com

A SIMULATION ALGORITHM FOR MULTIENVIRONMENT PROBABILISTIC P SYSTEMS: A FORMAL VERIFICATION

M. A. MARTÍNEZ-DEL-AMOR, I. PÉREZ-HURTADO,
M. J. PÉREZ-JIMÉNEZ, A. RISCOS-NÚÑEZ, F. SANCHO-CAPARRINI

Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla, Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
{*mdelamor,perezh,marper,ariscosn,fsancho*}*@us.es*
http://www.gcn.us.es

Received 21 June 2010
Accepted 1 October 2010
Communicated by Rudolf Freund

Multienvironment probabilistic P systems provide a framework of specification for modeling population biology. It has been used to model real ecosystems in a comprehensible, modular and probabilistic way. However, simulators are needed for virtual experimentation. Hence, the development of correct simulation algorithms becomes a critical point. In this paper we present a formal verification of a new algorithm of simulation designed for this kind of probabilistic P systems.

Keywords: P systems; biological modeling; formal verification.

1. Introduction

Modeling biological, chemical and physical phenomena has been, in the last years, a trend to analyze, describe and understand the intrinsic knowledge of these complex systems. Furthermore, computational formal models have been used as auxiliary tools for improving the development of experiments in many empirical sciences. In this sense, P systems have been used as a modeling framework for systems biology and population dynamics [1, 6]. P systems are theoretical computational devices defined in the field of Membrane Computing, first introduced by Gh. Păun in 1998 [5].

Recently, a P systems based general framework for modeling ecosystems dynamics was presented in [1]. This computational modeling framework has been employed for real ecosystems, such as the scavenger birds in the Catalan Pyrenees [2] and the zebra mussel in Ribarroja reservoir (located in Tarragona, Spain)[1]. It is noteworthy that the development of these models has been supervised by expert ecologists. The mentioned framework is based on multienvironment probabilistic P systems. The computations of these P systems behave non-deterministically, according to probabilistic functions that follow, as recommended by experts, the binomial distribution. Moreover, the modularity of these models also enables to easily adding or removing new ingredients and characteristics.

The aim of this P system based modeling framework is to help the ecologists to adopt *a priori* strategies in the real systems by executing virtual experiments. Therefore, the design of simulators and other related software tools becomes a critical point in the process of model validation, as well as for virtual experimentation. Thus, a software tool, based on pLinguaCore library [3], was developed and presented in [1].

In this paper, we present a formal verification of a new simulation algorithm called DNDP, in the sense that we prove that for each computation step, the multiset of rules selected to be applied by DNDP algorithm is maximally consistent (this notion will be explained later), and we also prove that DNDP correctly generates the next configuration obtained after the simulated step. This kind of formal checking is important in order to properly simulate the mentioned models based on probabilistic P systems.

The rest of the paper is structured as follows. Section 2 describes the modeling framework based on probabilistic P systems. Section 3 depicts the details of the DNDP simulation algorithm. In section 4 we show a formal verification of the algorithm. The paper ends with some conclusions and ideas for future work.

2. A Computational Modeling Framework

First, let us define the syntactical specifications of a P systems based framework for population biology, bringing in additional features, such as 3 electrical charges which turn out to be quite convenient for handling some specific features in a better way[a].

A multienvironment probabilistic functional P system with active membranes of degree (q, m) taking T time units can be viewed as a collection of m connected environments $e_1, \ldots, e_m$ (the structure of connections is given by the arcs from a directed graph) such that each environment e_j contains a probabilistic functional P system with active membranes of degree q, $\Pi_j = (\Gamma, \mu, R_\Pi, \{f_{r,j} : r \in R_\Pi\}, \{\mathcal{M}_{i,j} : 0 \leq i \leq q-1\})$. All Π_j, for $j = 1, \ldots, m$, share the *same skeleton*, (Γ, μ, R_Π); each rule of the system has associated a probabilistic function $f_{r,j}$; and the tuple $\mathcal{M}_{0,j}, \ldots, \mathcal{M}_{q-1,j}$ describes the initial multisets.

Definition 1. *A multienvironment probabilistic functional P system with active membranes of degree (q, m) with $q \geq 1$, $m \geq 1$, taking T time units, $T \geq 1$, is a tuple*

$$(G, \Gamma, \Sigma, T, \mu, R_E, R_\Pi, \{f_{r,j} : r \in R_\Pi, 1 \leq j \leq m\}, \{\mathcal{M}_{i,j} : 0 \leq i \leq q-1,\ 1 \leq j \leq m\})$$

where:

- *$G = (V, S)$ is a directed graph such that $(x, x) \in S$, for each $x \in V$. The elements of the set $V = \{e_1, \ldots, e_m\}$ are called environments;*
- *Γ is the working alphabet and $\Sigma \subsetneqq \Gamma$ is an alphabet representing the objects that can be present in the environments;*
- *T is a natural number that bounds the number of steps of the system. In some sense, it indicates the look-ahead time that we will work with in the modeling;*

[a] Details about the use of charges in the models are out of the scope of this paper, see e.g. [1].

- *μ is a membrane structure (that is, a rooted tree) consisting of q membranes injectively labeled with $0, 1, \ldots, q-1$. The skin membrane is labeled with 0. We also associate electrical charges with membranes from the set $\{0, +, -\}$;*
- *R_E is a finite set of communication rules between environments of the form:*

$$(x)_{e_j} \xrightarrow{p_{(x,j,j_1,\ldots,j_h)}} (y_1)_{e_{j_1}} \ldots (y_h)_{e_{j_h}}$$

  where $x, y_1, \ldots, y_h \in \Sigma$, $(e_j, e_{j_l}) \in S$ $(l = 1, \ldots, h)$ and $p_{(x,j,j_1,\ldots,j_h)}$ is a computable function ranging over $[0, 1]$ and whose domain is $\{1, 2, \ldots, T\}$. If $p_{(x,j,j_1,\ldots,j_h)}(t) = 1$, for each t, then we omit it. These rules verify that for each environment e_j and for each object $x \in \Sigma$, the sum of functions associated with the rules from R_E whose LHS (left–hand side) is $(x)_{e_j}$ coincide with the constant function equal to 1;
- *R_Π is a finite set of evolution rules of the form $r : u\ [\ v\ ]_i^\alpha \to u'\ [\ v'\ ]_i^{\alpha'}$ where u, v, u', v' are multisets over Γ, $i \in \{0, 1, \ldots, q-1\}$, and $\alpha, \alpha' \in \{0, +, -\}$;*
- *For each $r \in R_\Pi$ and for each j, $1 \leq j \leq m$, $f_{r,j}$ is a computable function such that $dom(f_{r,j}) \subseteq \{1, \ldots, T\}$, and $range(f_{r,j}) \subseteq [0, 1]$ verifying that $\sum_{h=1}^{z} f_{r_h,j}(t) = 1$, for each t, $1 \leq t \leq T$, where $r_1, \ldots, r_z$ are the rules from R_Π whose LHS is the same as r;*
- *For each i, $0 \leq i \leq q-1$ and for each j, $1 \leq j \leq m$, $\mathcal{M}_{i,j}$ are strings over Γ, describing the multisets of objects initially placed in the q regions of Π_j.*

The configuration of the system at any instant is a tuple $(C^1, \ldots, C^m)$ where C^j is the configuration of the environment e_j at that moment, that is, the tuple of multisets of objects present in the environment e_j and in the q *regions* of Π_j, together with their respective polarizations. The tuple $((\emptyset, \mathcal{M}_{0,1}, \ldots, \mathcal{M}_{q-1,1}), \ldots, (\emptyset, \mathcal{M}_{0,m}, \ldots, \mathcal{M}_{q-1,m}))$ with neutral polarizations in every membrane, is the initial configuration of the system.

Rules are applied as follows:

- When a communication rule $(x)_{e_j} \xrightarrow{p_{(x,j,j_1,\ldots,j_h)}} (y_1)_{e_{j_1}} \ldots (y_h)_{e_{j_h}}$ is applied, object x passes from e_j to $e_{j_1}, \ldots, e_{j_h}$ possibly modified into objects $y_1, \ldots, y_h$, respectively. In any moment t, $1 \leq t \leq T$, communication rules will be maximally applied to all available objects occurring in the LHS of some rule of R_E (the selection is done according to the probabilities of the rules).
- A rule $u[v]_i^\alpha \to u'[v']_i^{\alpha'}$ is applicable to a membrane labeled by i and with α as electrical charge, if multiset u is contained in the father of membrane i and multiset v is contained in the membrane labeled by i having α as electrical charge. When that rule is applied, multiset u (resp. v) in the father of membrane i (resp. in membrane i) is removed from that membrane, and the multiset u' (resp. v') is produced instead, setting the charge of membrane i to α'.
- The rules of each P system Π_j are applied in a *maximal consistent parallel* way; that is, a maximal multiset of applicable rules is selected (according to their associated probabilities), keeping in mind that for each $i \in \{0, 1, \ldots, q-1\}$, all applicable rules of the type $u[v]_i^\alpha \to u'[v']_i^{\alpha'}$ selected to be applied must agree on the charge α' of the RHS (consistence).

We assume that a global clock exists, marking the time for the whole system (for its compartments), that is, all membranes and the application of all rules are synchronized.

3. The DNDP Simulation Algorithm

In this section we describe the design and the pseudocode of the DNDP (Direct Non-deterministic Distribution with Probabilities) simulation algorithm. Its aim is to overcome the disadvantages of the algorithm introduced in [1], that restricted the variety of P systems models that could be correctly simulated. The input is a multienvironment probabilistic functional extended P system with active membranes of degree (q, m), taking T time units. The algorithm simulates only one computation of the P system (actually, only T transition steps), by executing rules in a non-deterministic maximal consistent parallel way.

Next we show the pseudocode of the DNDP algorithm.

Input: A multienvironment functional P system with active membranes of degree (q, m) with $q \geq 1,\ m \geq 1$, taking T time units, $T \geq 1$.

1: $C_0 \leftarrow$ initial configuration of the system
2: **for** $t \leftarrow 0$ to $T - 1$ **do**
3: $\quad C'_t \leftarrow C_t$
4: $\quad$ *Initialization*
5: $\quad$ *First selection phase*: generates a multiset of *consistent* applicable rules.
6: $\quad$ *Second selection phase*: generates a multiset of *maximal consistent* applicable rules.
7: $\quad$ *Execution of selected rules*.
8: $\quad C_{t+1} \leftarrow C'_t$
9: **end for**

Roughly speaking, the simulation of each transition step is divided into two phases: *selection* and *execution*. In the first one, a multiset of maximally consistent applicable rules is calculated. In the last one, the rules in the multiset are applied to the configuration, each one as many times as indicated by their multiplicity. Note that adding new objects before finishing selection phase could mislead the algorithm yielding inconsistent states, since the algorithm could use such new objects for triggering rules of the P system that were not supposed to be applied until the next transition step. Cooperation is another feature to be taken into account, in the sense that left-hand sides of the rules consist of several objects "cooperating" to activate the rule. Moreover, if there are rules with overlapping left-hand sides, then they will compete for the common objects. Therefore, the algorithm should efficiently perform an object distribution.

Finally, execution phase will add the RHS (right-hand side) of the rules in the maximal consistent applicable multiset of rules to the configuration C'_t. At the end of the process, C'_t is actually the next configuration C_{t+1}: the LHS of rules has been removed in the first and second selection phases, and the RHS of rules is added in the execution stage.

Let us now describe the pseudocode of the four main modules of the algorithm.

First of all, in order to simplify the selection and execution phases, the *initialization* process constructs two ordered set of rules, A_j and B_j, gathering only rules from R_E and

R_Π applicable in environment e_j, and having a probability greater than 0.

Initialization

1: $R_\Pi \leftarrow$ ordered set of rules of Π
2: **for** $j \leftarrow 1$ to m **do**
3: $R_{E,j} \leftarrow$ ordered set of rules from R_E related to the environment j
4: $A_j \leftarrow$ ordered set of rules from $R_{E,j}$ whose probability at the moment t is > 0
5: $LC_j \leftarrow$ ordered set of pairs $\langle label, charge \rangle$ for all the membranes from C_t contained in the environment j
6: $B_j \leftarrow \emptyset$
7: **for** each $\langle h, \alpha \rangle \in LC_j$ (following the considered order) **do**
8: $B_j \leftarrow B_j \cup$ ordered set of rules $u[v]_h^\alpha \rightarrow u'[v']_h^\beta$ from R_Π whose probability at the moment t is greater than 0 for the environment j
9: **end for**
10: **end for**

The selection process is split into two phases, following the design of the DND (Direct Non-deterministic Distribution) algorithm introduced by Nguyen et al. in [4]. The *first selection* phase generates a multiset of consistent applicable rules, where the number of times each rule will be applied is randomly calculated by running a binomial distribution according to their associated probability function, as explained below. The *second selection* phase eventually increases the multiplicity of some of these rules, obtaining a multiset of maximal consistent applicable rules.

First selection phase (*consistence*)

1: **for** $j \leftarrow 1$ to m **do**
2: $R_j \leftarrow$ *the empty multiset*
3: $D_j \leftarrow A_j \cup B_j$ with a *random order*
4: **for** each $r \in D_j$ (following the considered order) **do**
5: $M \leftarrow$ maximum number of times that r is applicable to C'_t
6: **if** r is *consistent* with the rules in $R_j^1 \wedge M > 0$ **then**
7: $N \leftarrow$ maximum number of times that r is applicable to C_t
8: $n \leftarrow \min\{M, F_b(N, f_{r,j}(t))\}$
9: $C'_t \leftarrow C'_t - n \cdot LHS(r)$
10: $R_j \leftarrow R_j \cup \{\langle r, n \rangle\}$
11: **end if**
12: **end for**
13: **end for**

In the first selection phase, a multiset of consistent applicable rules, denoted by R_j for each environment j, is calculated. First, a random order is applied to $A_j \cup B_j$, and stored in an ordered set D_j. Moreover, a copy of the configuration C_t, called C'_t, is created and it is updated each time that a rule is selected (removing the LHS).

Then, a rule r is applicable if the following holds: it is consistent with the previously[b] selected rules in R_j, and the number of possible applications M in C'_t is greater than 0. On the one hand, since C'_t has been updated by the previously selected rules, the number

[b] According to the order in D_j.

n cannot exceed M to guarantee a correct object distribution. On the other hand, if the generated number n is 0, the corresponding rule is also added to the multiset R_j, giving a new chance to be selected in the next phase (maximality). Therefore, we will handle the "multiset" of rules R_j as a set of pairs $\langle x, y\rangle$ where $x \in D_j$ and y is the number of times that x is going to be applied (eventually $y = 0$). We will denote $R_j^0 = \{r \in D_j : \langle r, 0\rangle \in R_j\}$ and $R_j^1 = \{r \in D_j : \langle r, n\rangle \in R_j, n > 0\}$.

In the second selection phase, rules from R_j are checked again in order to achieve maximality. If one rule $r \in R_j$ has a number of applications N greater than 0 in C'_t, N will be directly assigned to the rule r. Rules from R_j are iterated in order according to the probabilities for fair object distribution. Note that rules in R_j^0 are also checked, so the condition of consistence has to be tested again.

Second phase of rules selection (*maximality*)

for $j \leftarrow 1$ to m **do**
  $R_j \leftarrow R_j$ *with an order by the rule probabilities, from highest to lowest*
  for each $\langle r, n\rangle \in R_j$ (following the selected order) **do**
    if $n > 0 \vee (r$ is *consistent* with the rules in $R_j^1)$ **then**
      $M \leftarrow$ maximum number of times that r is applicable to C'_t
      if $M > 0$ **then**
        $R_j \leftarrow R_j \cup \{\langle r, M\rangle\}$
        $C'_t \leftarrow C'_t - M \cdot LHS(r)$
      end if
    end if
  end for
end for

In order to complete the simulation of the computation step, the last phase (execution) takes care of the effects of applying the rules selected in the previous phase: updating the charges according to the RHS of the rules and adding the necessary objects.

Execution of selected rules

for $j \leftarrow 1$ to m **do**
  for each $\langle r, n\rangle \in R_j$ with $n > 0$ **do**
    $C'_t \leftarrow C'_t + n \cdot RHS(r)$
    Update the electrical charges of C'_t according to $RHS(r)$
  end for
end for

4. A Formal Verification

Let us start by analyzing the first selection phase (*consistence*). Its goal is to select a multiset of applicable rules for C_t, trying to capture the stochasticity of the system.

4.1. *Verification of the first selection phase (consistence)*

This module receives as input:

- A number t ($0 \leq t \leq T - 1$), that indicates the step of the computation that is

being simulated (actually, step t of the main loop of DNDP algorithm refers to the $(t+1)$-th step of the P system computation).
- A number j ($1 \leq j \leq m$), representing which environment is being considered.
- C_t, the configuration at time t of the simulated multienvironment functional probabilistic P system with active membranes of degree (q, m).
- The set A_j of all rules from R_E applicable on environment e_j and having a probability at time t strictly greater than 0.
- The set B_j of all rules $r \in R_\Pi$ applicable on Π_j such that their probability is strictly greater than 0.

The output generated by this module is a multiset of rules R_j and a configuration C'_t.

The following theorems formalize the concept of correctness that we want to prove for this module:

Theorem 2. *The multiset of rules R_j is applicable to C_t, and the result of removing the objects consumed by those rules is C'_t.*

Theorem 3. *There exists a maximally consistent multiset of applicable rules for C_t that can be obtained from R_j. That is, any rule that does not appear in R_j, cannot be consistently applied to C'_t.*

Before proceeding with the details of the formal verification, let us re-label the module.

1: $C'_{t,0} \leftarrow C_t$
2: $R_{j,0} \leftarrow \emptyset$
3: $D_j \leftarrow A_j \cup B_j = \{r^j_1, \ldots, r^j_{s_j}\}$ with a *random order* ($r^j_1 < \cdots < r^j_{s_j}$)
4: **for** $\alpha \leftarrow 1$ to s_j **do**
5: $\quad M^j_\alpha \leftarrow$ maximum number of times that r^j_α is applicable to $C'_{t,\alpha-1}$
6: $\quad$ **if** r^j_α is *consistent* with the rules in $R^1_{j,\alpha-1} \wedge M^j_\alpha > 0$ **then**
7: $\quad\quad N^j_\alpha \leftarrow$ maximum number of times that r^j_α is applicable to C_t
8: $\quad\quad n^j_\alpha \leftarrow \min\{M^j_\alpha, F_b(N^j_\alpha, f_{r^j_\alpha,j}(t))\}$
9: $\quad\quad C'_{t,\alpha} \leftarrow C'_{t,\alpha-1} - n^j_\alpha \cdot LHS(r^j_\alpha)$
10: $\quad\quad R_{j,\alpha} \leftarrow R_{j,\alpha-1} \cup \{\langle r^j_\alpha, n^j_\alpha\rangle\}$
11: $\quad$ **else**
12: $\quad\quad C'_{t,\alpha} \leftarrow C'_{t,\alpha-1}$
13: $\quad\quad R_{j,\alpha} \leftarrow R_{j,\alpha-1}$
14: $\quad$ **end if**
15: **end for**

where $f_{r^j_\alpha,j}(t)$ stands for the probability of applying rule r^j_α at time t.

Remark 4. *We shall use the following notation.*

- $l(R_{j,h}) = \{r : \exists n\ (\langle r, n\rangle \in R_{j,h})\}$
- $R^1_{j,h} = \{r : \exists n > 0\ (\langle r, n\rangle \in R_{j,h})\}$
- $M^j_h =$ *maximum number of times that rule r^j_h can be applied to configuration $C'_{t,h-1}$.*

- $N_h^j =$ *maximum number of times that rule* r_h^j *can be applied to configuration* C_t.

Let us consider the following formula $\varphi(\alpha)$, for any α $(\alpha = 1, \ldots, s_j)$.

$$\begin{aligned}
&\forall \beta\ (1 \leq \beta \leq \alpha \implies \\
&\quad [r_\beta^j \in D_j - l(R_{j,\alpha}) \implies (r_\beta^j \text{ is not consistent with } R_{j,\alpha-1}^1) \vee (M_\beta^j = 0)] \wedge \\
&\quad [\langle r_\beta^j, n_\beta^j \rangle \in R_{j,\alpha} \implies (r_\beta^j \text{ is consistent with } R_{j,\alpha-1}^1) \wedge M_\beta^j > 0 \wedge n_\beta^j \leq M_\beta^j \wedge \\
&\qquad (C'_{t,\beta} = C'_{t,\beta-1} - n_\beta^j \cdot LHS(r_\beta^j)) \wedge R_{j,\beta} = R_{j,\beta-1} \cup \{\langle r_\beta^j, n_\beta^j \rangle\}])
\end{aligned}$$

Theorem 5. $\forall \alpha\ (1 \leq \alpha \leq s_j) \implies \varphi(\alpha)$.

Proof. By induction on α. For the base case, let $r_1^j \in D_j$. In this case, $R_{j,0}^1 = R_{j,0}^0 = \emptyset$. Thus, r_1^j is clearly consistent with $R_{j,0}^1$. Moreover, $N_1^j \geq M_1^j \geq 0$.

- If $M_1^j = 0$ then we have $C'_{t,1} = C'_{t,0} = C_t$, and $R_{j,1} = R_{j,0} = \emptyset$
- If $M_1^j > 0$ then $n_1^j = \min\{F_b(N_1^j, f_{r_1^j,j}), M_1^j\}$, $C'_{t,1} = C'_{t,0} - n_1^j \cdot LHS(r_1^j)$, and $R_{j,1} = \{\langle r_1^j, n_1^j \rangle\}$

Thus, $\varphi(1)$ follows.

Let α be such that $1 \leq \alpha < s_j$ and let us suppose that the formula $\varphi(\alpha)$ holds. Let β be such that $1 \leq \beta \leq \alpha + 1$.

Let us suppose $1 \leq \beta \leq \alpha$. If $r_\beta^j \in D_j - l(R_{j,\alpha+1})$, then $r_\beta^j \in D_j - l(R_{j,\alpha})$ and, by applying the induction hypothesis, we have either r_β^j is not consistent with $R_{j,\beta-1}^1$ or r_β^j is consistent with $R_{j,\beta-1}^1$ and $M_\beta^j = 0$. If $r_\beta^j \in l(R_{j,\alpha+1})$ then

- Case 1: $r_\beta^j \in l(R_{j,\alpha})$. In this case all properties are deduced from the veracity of $\varphi(\alpha)$.
- Case 2: $r_\beta^j \in l(R_{j,\alpha+1}) - l(R_{j,\alpha})$. In this case, $\langle r_\beta^j, n_\beta^j \rangle$ should have been added to the set $R_{j,\alpha+1}$ at the step $\alpha + 1$. But this is not possible because $1 \leq \beta \leq \alpha$.

Let us suppose now that $\beta = \alpha + 1$. Then, we distinguish two cases.

- Case 1: $r_{\alpha+1}^j \in D_j - l(R_{j,\alpha+1})$ and $r_{\alpha+1}^j$ is consistent with $R_{j,\alpha}^1$. Then $M_{\alpha+1}^j = 0$ because on the contrary we have $r_{\alpha+1}^j \in l(R_{j,\alpha+1})$ according to the semantics of the algorithm (more precisely, we have $R_{j,\alpha+1} = R_{j,\alpha} \cup \{\langle r_{\alpha+1}^j, n_{\alpha+1}^j \rangle\}$).
- Case 2: $r_{\alpha+1}^j \in l(R_{j,\alpha+1})$. In this case, that rule has been added to the set $l(R_{j,\alpha+1})$ because $r_{\alpha+1}^j$ is consistent with $R_{j,\alpha}^1$ and $M_\alpha^j > 0$. Moreover, we have: $n_{\alpha+1}^j = \min\{F_b(N_{\alpha+1}^j, f_{r_{\alpha+1}^j,j}), M_{\alpha+1}^j\}$, $C'_{t,\alpha+1} = C'_{t,\alpha} - n_{\alpha+1}^j \cdot LHS(r_{\alpha+1}^j)$, and $R_{j,\alpha+1} = R_{j,\alpha} \cup \{\langle r_{\alpha+1}^j, n_{\alpha+1}^j \rangle\}$ □

The previous theorem implies that, in particular, the formula $\varphi(s_j)$ holds.

Let us denote $R_{j,s_j} = \{\langle r_{\delta_1}, n_{\delta_1} \rangle, \ldots, \langle r_{\delta_k}, n_{\delta_k} \rangle\}$, where $1 \leq \delta_1 < \cdots < \delta_k \leq s_j$

Proposition 6. *The set of rules $\{r_{\delta_1}, \ldots, r_{\delta_k}\}$ is consistent, that is, for any pair of rules, if their LHS refer to the same membrane label and charge (e.g. $u\ [\ v\ ]_h^\alpha$ and $u'\ [\ v'\ ]_h^\alpha$), then both RHS must agree on the new charge (e.g. $u''\ [\ v''\ ]_h^{\alpha'}$ and $u'''\ [\ v'''\ ]_h^{\alpha'}$).*

Proof. Since the formula $\varphi(s_j)$ holds, we deduce

$$\forall \beta\ ((1 \leq \beta \leq s_j \wedge r_\beta \in l(R_{j,s_j})) \implies r_\beta \text{ is consistent with } R^1_{j,\beta-1})$$

Hence, $\forall i\ (1 \leq i \leq k \implies r_{\delta_i}$ is consistent with $\{r_{\delta_1}, \ldots, r_{\delta_k}\})$ □

Proposition 7. *For all i, $1 \leq i \leq k$ we have:*

$$C'_{t,\delta_i} = C'_{t,\delta_{i-1}} - n_{\delta_i} \cdot LHS(r_{\delta_i}) \wedge \forall \gamma\ (\delta_{i-1} < \gamma < \delta_i \implies C'_{t,\gamma} = C'_{t,\delta_{i-1}})$$

Proof. (Sketch) The proof can be easily deduced from Theorem 5, taking into account that $r_{\delta_i} \in l(R_{j,s_j})$, for every i, $1 \leq i \leq k$, and $\forall \gamma\ (\delta_{i-1} < \gamma < \delta_i \implies r_\gamma \notin l(R_{j,s_j}))$. □

Corollary 8. *For all i, $1 \leq i \leq k$ we have $C'_{t,\delta_i} = C_t - \bigcup_{h=1}^{i} n_{\delta_h} \cdot LHS(r_{\delta_h})$.*

Proof. (Sketch) The proof follows from proposition 7 reasoning by induction. □

Corollary 9. $C'_{t,s_j} = C'_{t,\delta_k} = C_t - \bigcup_{r \in R_{j,s_j}} n_r \cdot LHS(r)$.

Proof. From Corollary 8, we have $C'_{t,\delta_k} = C_t - \bigcup_{r \in R_{j,s_j}} n_r \cdot LHS(r)$. Moreover, $\forall \gamma\ (\delta_k < \gamma \leq s_j \implies C'_{t,\gamma} = C'_{t,\delta_k})$. Indeed, let γ be such that $\delta_k < \gamma \leq s_j$, then $\gamma \in D_j - l(R_{j,s_j})$. Having in mind the formula $\varphi(s_j)$ is true we deduce that rule r_γ is not applied at the current configuration at the step γ. Hence, $C'_{t,\gamma} = C'_{t,\delta_k}$. □

Corollary 10. *The multiset of rules R_{j,s_j} is applicable in a consistent manner to the configuration C_t.*

Proof. It is enough to notice that for all i, $1 \leq i \leq q$, the rule r_{δ_i} is applicable to $C'_{t,\delta_{i-1}}$ and consistence is always checked before adding a new pair $\langle r_{\delta_i}, n_{\delta_i} \rangle$ to R_{j,δ_i}. □

Note that the proof of Theorem 2 follows from Corollary 9 and Corollary 10.

Proposition 11. *If $r \in D_j - l(R_{j,s_j})$ then the rule r is not applicable in a consistent manner to the configuration C'_{t,s_j}.*

Proof. From Corollary 9 we have $C'_{t,s_j} = C_t - \bigcup_{r \in R_{j,s_j}} n_r \cdot LHS(r)$. Since formula $\varphi(s_j)$ holds, we deduce that if $r_\beta \in D_j - l(R_{j,s_j})$ then either r_β is not consistent with $R^1_{j,\beta-1}$, or r_β is consistent with $R^1_{j,\beta-1}$ and $M^j_\beta = 0$. □

From this last result we deduce that in order to determine a maximal consistent multiset of applicable rules for a configuration C_t, it suffices to take rules from the set $l(R_{j,s_j})$ and apply them in a maximal way. That is, we deduce that Theorem 3 holds.

4.2. *Verification of the second phase of rules selection (consistent maximality)*

This module receives as input two numbers t and j ($0 \leq t \leq T-1$ and $1 \leq j \leq m$), like in the previous module, and it also receives the following:

- A multiset of rules $R_{j,s_j} = \{\langle r_{\gamma_1}, n_{\gamma_1}\rangle, \ldots, \langle r_{\gamma_k}, n_{\gamma_k}\rangle\}$, obtained as output of the previous algorithmic module, with an order given by the probabilities of the rules at time t, from highest to lowest.
- An intermediate configuration C'_t, obtained from configuration C_t by removing all objects consumed by the application of the multiset R_{j,s_j}.

The output generated by this module is a multiset of rules R_j and a configuration C'_t.

The following theorems formalize the concept of correctness that we want to prove for this module:

Theorem 12. *The multiset of rules R^1_j is maximally applicable in a consistent manner to the configuration C_t.*

Theorem 13. *The configuration C'_t obtained as output of the module is the result of removing from C_t the objects consumed by the application of the multiset of rules R^1_j.*

Before proceeding with the formal verification, let us re-label the algorithmic module.

1: **for** $i \leftarrow 1$ to k **do**
2: **if** $n_{\gamma_i} > 0 \vee (r_{\gamma_i}$ is *consistent* with $R^1_{sel,j})$ **then**
3: $M^j_{s_j+i} \leftarrow \max\{$number of times that r_{γ_i} is applicable to $C'_{t,s_j+i-1}\}$
4: **if** $M^j_i > 0$ **then**
5: $R^1_{j,s_j+i} \leftarrow R^1_{j,s_j+i-1} \cup \{\langle r_{\gamma_i}, M^j_{s_j+i}\rangle\}$
6: $C'_{t,s_j+i} \leftarrow C'_{t,s_j+i-1} - M^j_{s_j+i} \cdot LHS(r_{\gamma_i})$
7: **else**
8: $R^1_{j,s_j+i} \leftarrow R^1_{j,s_j+i-1}$
9: $C'_{t,s_j+i} \leftarrow C'_{t,s_j+i-1}$
10: **end if**
11: **end if**
12: **end for**

From the design of the algorithm we deduce this result.

Proposition 14. *The following formula $\psi(i)$ is true for $1 \leq i \leq k$:*

$$\begin{aligned}
&\big[n_{\gamma_i} > 0 \wedge M^j_{s_j+i} > 0 \wedge (r_{\gamma_i} \textit{ is applicable } M^j_{s_j+i} \textit{ times to } C'_{t,s_j+i-1} \textit{ but not } M^j_{s_j+i}) \\
&\qquad\qquad +1 \textit{ times} \wedge\ (C'_{t,s_j+i} = C'_{t,s_j+i-1} - M^j_{s_j+i} \cdot LHS(r_{\gamma_i})) \\
&\qquad\qquad \wedge (R_{j,s_j+i} = R_{j,s_j+i-1} \cup \langle r_{\gamma_i}, M^j_{s_j+i}\rangle)\big] \vee \\
&\big[n_{\gamma_i} > 0 \wedge M^j_{s_j+i} = 0 \wedge (C'_{t,s_j+i} = C'_{t,s_j+i-1})\ \wedge\ (R_{j,s_j+i} = R_{j,s_j+i-1})\big] \vee \\
&\big[n_{\gamma_i} = 0 \wedge M^j_{s_j+i} > 0 \wedge (r_{\gamma_i} \textit{ is consistent with } R^1_{j,s_j+i-1}) \\
&\qquad\qquad \wedge (C'_{t,s_j+i} = C'_{t,s_j+i-1} - M^j_{s_j+i} \cdot LHS(r_{\gamma_i})) \\
&\qquad\qquad \wedge (R_{j,s_j+i} = R_{j,s_j+i-1} \cup \langle r_{\gamma_i}, M^j_{s_j+i}\rangle)\big]
\end{aligned}$$

Proposition 15. *The multiset $R^1_{j,s_j+k} = \{\langle r_{\gamma_1}, n_{\gamma_1} + M^j_{s_j+1}\rangle, \ldots, \langle r_{\gamma_k}, n_{\gamma_k} + M^j_{s_j+k}\rangle\}$ is a multiset of rules maximal consistent for the configuration C_t. Moreover, C'_{t,s_j+k} is the configuration obtained from C_t removing the objects consumed by the application of the multiset of rules R^1_{j,s_j+k}.*

Proof. The maximality of the multiset R^1_{j,s_j+k} follows from the construction of the multiplicities $M^j_{s_j+i}$. Let us recall that $C'_{t,s_j} = C_t - n_{\gamma_1} \cdot LHS(r_{\gamma_1}) - \cdots - n_{\gamma_k} \cdot LHS(r_{\gamma_k})$. From Proposition 14 we deduce that:

$$\begin{cases}
C'_{t,s_j+1} = C'_{t,s_j} - M^j_{s_j+1} \cdot LHS(r_{\gamma_1}) \\
C'_{t,s_j+2} = C'_{t,s_j+1} - M^j_{s_j+2} \cdot LHS(r_{\gamma_2}) \\
\cdots \qquad\qquad \cdots \\
C'_{t,s_j+k} = C'_{t,s_j+k-1} - M^j_{s_j+k} \cdot LHS(r_{\gamma_k})
\end{cases}$$

Note that this proposition corresponds to a re-labelling of Theorem 12 and Theorem 13. Thus, the correctness of the module follows. □

4.3. *Execution of selected rules*

This module receives as input, for each j ($1 \leq j \leq m$), the configuration C'_{t,s_j+q} and the selected multiset of rules $R^1_{j,s_j+k} = \{\langle r_{\gamma_1}, n_{\gamma_1} + M^j_{s_j+1}\rangle, \ldots, \langle r_{\gamma_k}, n_{\gamma_k} + M^j_{s_j+k}\rangle\}$. The output is C'_{t,s_j+2k} which is the next configuration C_{t+1}.

1: **for** $j \leftarrow 1$ to m **do**
2: **for** $i \leftarrow 1$ to k **do**
3: $C'_{t,s_j+k+i} \leftarrow C'_{t,s_j+k+i-1} + (n_{\gamma_i} + M^j_{s_j+i}) \cdot RHS(r_{\gamma_i})$
4: Update the electrical charges of C'_{t,s_j+k+i} according to $RHS(r_{\gamma_i})$
5: **end for**
6: **end for**

From Proposition 15 we deduce that C'_{t,s_j+k} is the configuration obtained from C_t removing the objects consumed by the application of the multiset of rules R^1_{j,s_j+k}. But configuration C'_{t,s_j+2k} is obtained from C'_{t,s_j+k} adding the objects produced by the application of multiset of selected rules R^1_{j,s_j+k} (and updating the corresponding polarizations). Thus, C'_{t,s_j+2k} is the configuration of the environment e_j obtained from C_t after the application of the multiset of selected rules R^1_{j,s_j+k}.

5. Conclusions

The correctness (in the sense explained in Section 4) of a new simulation algorithm for multienvironment probabilistic functional extended P systems with active membranes has been demonstrated in this paper. This new algorithm, called DNDP, has been presented and formally verified. The framework for modeling population dynamics based on multienvironment probabilistic P systems has been also described.

Finally, correct simulation algorithms are needed to develop simulators, which are required for model validation process and virtual experimentation. In this sense, new simulators and software tools, based on both sequential and parallel platforms, are under development considering the correctness of the algorithm.

Another interesting issue is to study the way in which objects are assigned to rules according to their probability. We do not deal with this in this paper, but it could be experimentally investigated in cooperation with expert ecologists.

Acknowledgments

The authors acknowledge the support of the project TIN2009–13192 of the Ministerio de Educación y Ciencia of Spain, cofinanced by FEDER funds, and the support of the Project of Excellence with *Investigador de Reconocida Valía* of the Junta de Andalucía, grant P08-TIC-04200.

References

[1] M. Cardona, M.A. Colomer, A. Margalida, A. Palau, I. Pérez-Hurtado, M.J. Pérez-Jiménez, D. Sanuy. A computational modeling for real ecosystems based on P systems, Natural Computing, online version (DOI: 10.1007/s11047-010-9191-3).

[2] M. Cardona, M.A. Colomer, A. Margalida, I. Pérez-Hurtado, M.J. Pérez-Jiménez, D. Sanuy. A P system based model of an ecosystem of some scavenger birds, LNCS 5957 (2010), 182–195.

[3] M. García-Quismondo, R. Gutiérrez-Escudero, I. Pérez-Hurtado, M.J. Pérez-Jiménez, A. Riscos-Núñez. An overview of P-Lingua 2.0, LNCS 5957 (2010), 264–288.

[4] V. Nguyen, D. Kearney, G. Gioiosa. An algorithm for non-deterministic object distribution in P systems and its implementation in hardware, LNCS 5391 (2009), 325–354.

[5] G. Păun. Computing with membranes. Journal of Computer and System Sciences, 61, 1 (2000), pp. 108–143, and Turku Center for Computer Science-TUCS Report No 208.

[6] Gh. Păun, F.J. Romero-Campero. Membrane Computing as a Modeling Framework. Cellular Systems Case Studies, LNCS 5016 (2008), 168–214.