ON A PARTIAL AFFIRMATIVE ANSWER FOR A PĂUN'S CONJECTURE

IGNACIO PÉREZ-HURTADO, MARIO J. PÉREZ-JIMÉNEZ,
AGUSTÍN RISCOS-NÚÑEZ and MIGUEL A. GUTIÉRREZ-NARANJO

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

MIQUEL RIUS-FONT

Department of Applied Mathematics IV, Universitat Politécnica de Catalunya
Edifici C3, Despatx 016, Av. del Canal Olímpic, s/n, 08860 Castelldefels, Spain
mrius@ma4.upc.edu

Received 21 June 2010
Accepted 30 September 2010
Communicated by Victor Mitrana

At the beginning of 2005, Gheorghe Păun formulated a conjecture stating that in the framework of recognizer P systems with active membranes (evolution rules, communication rules, dissolution rules and division rules for elementary membranes), polarizations cannot be avoided in order to solve computationally hard problems efficiently (assuming that **P** $\neq$ **NP**). At the middle of 2005, a partial positive answer was given, proving that the conjecture holds if dissolution rules are forbidden. In this paper we give a detailed and complete proof of this result modifying slightly the notion of dependency graph associated with recognizer P systems.

Keywords: Polarizationless P systems; active membranes; dissolution rules; computational complexity; tractability.

1. Preliminaries

The present work is developed within the framework of polarizationless P systems with active membranes where division rules can only be applied to elementary membranes.

In order to make this paper selfcontained, let us introduce next the required concepts.

1.1. *Notations*

Let us recall that an alphabet Γ is a nonempty set and a string on Γ is a finite sequence of elements of Γ. The set of symbols ocurring in a string w is denoted by $alph(w)$. We denote by Γ^* the set of all strings over Γ. A language over Γ is a subset of Γ^*.

A multiset over an alphabet Γ is a map from Γ into the set of natural numbers, **N**. The support of a multiset f over Γ is $supp(f) = \{a \in \Gamma : \; f(a) > 0\}$. If f, g are multisets over Γ, then we define the union or sum of f and g as follows: the multiset $f + g$ defined by $(f + g)(a) = f(a) + g(a)$, for each $a \in \Gamma$.

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 (that is, a predicate) over I_X. Given a decision problem $X = (I_X, \theta_X)$ we can associate a language to it as follows: $L_X = \{w \in I_X : \; \theta_X(w) = 1\}$. Conversely, given a language L, over an alphabet Γ, we can associate a decision problem to it as follows: $X_L = (I_{X_L}, \theta_{X_L})$, where $I_{X_L} = \Gamma^*$, and $\theta_{X_L} = \{(x, 1) : \; x \in L\} \cup \{(x, 0) : \; x \notin L\}$.

In this paper we work only with finite alphabets.

1.2. *The graph reachability problem and the circuit value problem*

The *graph reachability* (or accessibility) *problem* is the following: given a directed graph, $G = (V, E)$, with two specified vertices s and t, determine whether or not there is a path from s to t. We denote this decision problem by REACHABILITY.

There are algorithms solving this problem, for instance, search algorithms based on breadth first search or depth-first search. These algorithms determine whether two vertices are connected in $O(\max(|V|, |E|))$ time. Moreover, they basically need to store at most $|V|$ items, so these algoritms use $O(|V|)$ space. But this quantity of space can be reduced to $O(\log^2 |V|)$ by using an algorithm that could be called middle-first search (see [2] for details, pp. 149-150). In particular, REACHABILITY $\in \mathbf{P}$.

The *circuit value problem* is the following: given a circuit with no variable gates and where each input gate has an associated boolean value, determine whether or not the circuit evaluates to *True*. It is well known that the circuit value problem is a **P**–complete problem (see theorem 8.1 in [2] for details).

1.3. *Polarizationless P system with active membranes*

Definition 1. *A* polarizationless P system with active membranes *of degree* $q \geq 1$ *is a tuple* $\Pi = (\Gamma, H, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, R, h_{output})$*, where:*

(1) Γ *is a working alphabet of objects, and* H *is a finite set of labels for membranes;*
(2) μ *is a membrane structure (a rooted tree) consisting of* q *membranes injectively labeled with elements of* H*, the label of membrane* i *is denoted by* $l(i)$*;*
(3) $\mathcal{M}_1, \ldots, \mathcal{M}_q$ *are strings over* Γ *describing the multisets of objects placed in the* q *initial regions of* μ*;*
(4) R *is a finite set of developmental rules, of the following forms:*

(a) $[\, a \to u \,]_h$*, for* $h \in H$*,* $a \in \Gamma$*,* $u \in \Gamma^*$ (object evolution rules).
(b) $a\, [\;\;]_h \to [\, b \,]_h$*, for* $h \in H$*,* $a, b \in \Gamma$ (send–in communication rules).
(c) $[\, a \,]_h \to [\;]_h\; b$*, for* $h \in H$*,* $a, b \in \Gamma$ (send–out communication rules).
(d) $[\, a \,]_h \to b$*, for* $h \in H$*,* $a, b \in \Gamma$ (dissolution rules).

(e) $[\,a\,]_h \to [\,b\,]_h\ [\,c\,]_h$, *for* $h \in H$, $a, b, c \in \Gamma$ (division rules for elementary membranes.

(5) $h_{output} \in H$ *or* $h_{output} = env$ *(*env *is the label of the environment) indicates the output region.*

These rules are applied according to usual principles of polarizationless P systems (see [4] for details). Notice that in this polarizationless framework there is no cooperation, nor priority, nor changes of the labels of membranes. In this paper we do not consider division rules for non-elementary membranes.

A membrane structure is a rooted tree. Thus, we can define the notion of the *father* $f(h)$ or a *child* $ch(h)$ of a region labeled by h in a natural manner. The *length* of a rule is the number of symbols ocurring in it. For instance, communication and dissolution rules have length 2, but division rules have length 3.

In order to solve decision problems in the so-called *uniform* way we will work with P systems with input membrane. These systems have an input alphabet Σ strictly contained in the working alphabet Γ, the initial multisets are over $\Gamma \setminus \Sigma$, and they also have a distinguished membrane (the input membrane) in which the instances of a problem are encoded by multisets over the input alphabet Σ. Specifically, if $\Pi = (\Gamma, \Sigma, H, \mu, \mathcal{M}_1, \dots, \mathcal{M}_q, R, h_{output}, h_{input})$, and $IM \in \Sigma^*$ then the initial configuration of Π with input IM is $(\mathcal{M}_1, \dots, \mathcal{M}_{input} + IM, \dots, \mathcal{M}_q)$. Usually, we write $\Pi + IM$.

Let us recall that in *recognizer P systems,* the working alphabet contains two distinguished elements *yes* and *no*, the output of the system is collected in the environment, all computations halt, and if $\mathcal{C}$ is a computation of the system, then either object *yes* or object *no* (but not both) must have been sent to the output region of the system, and only at the last step of the computation.

Without loss of generality, we can suppose that in a recognizer P system with active membranes Π, for each label $h \in H$ the number of rules associated with membrane h is lower than or equal to $2 \cdot |\Gamma|$ (the non-determinism among all the rules associated with a given membrane, except those of send-in type, can be avoided). Thus, we can suppose that $|R_\Pi| \leq 2q \cdot |\Gamma|$, where q is the degree of Π.

The class of recognizer polarizationless P systems with active membranes (resp., which do not make use of division rules) is denoted by $\mathcal{AM}^0$ (resp., $\mathcal{NAM}^0$). If we consider three electrical charges the corresponding classes are denoted by $\mathcal{AM}$ and $\mathcal{NAM}$, respectively. We also denote by $\mathcal{AM}^0(-d)$ the class of all recognizer P systems with polarizationless active membranes such that dissolution rules are forbidden.

Definition 2. *A decision problem* $X = (I_X, \theta_X)$ *is* solvable in polynomial time *by a family of recognizer P systems with input membrane* $\mathbf{\Pi} = \{\Pi(n) : n \in \mathbf{N}\}$*, denoted by* $X \in \mathbf{PMC}_{\mathcal{R}}$*, if the following holds:*

- *There exists a deterministic Turing machine (DTM) working in polynomial time which constructs the system* $\Pi(n)$ *from* $n \in \mathbf{N}$*.*
- *There exists a pair* (cod, s) *of polynomial time computable functions over* I_X *(a polynomial encoding of* X *in* $\mathbf{\Pi}$*) such that for each instance* $w \in I_X$*,* $s(w)$ *is a natural*

number (obtained by means of a reasonable encoding scheme) and $cod(w)$ *is an input multiset of the system* $\Pi(s(w))$.

- *There exists* $k \in \mathbf{N}$ *such that for each instance* $w \in I_X$, *every computation of the system* $\Pi(s(w))$ *with input* $cod(w)$ *performs at most* $|w|^k$ *steps.*
- *For each instance,* $w \in I_X$*: (a) if* there exists *an accepting computation of* $\Pi(w)$, *then* $\theta_X(w) = 1$*; and (b) if* $\theta_X(w) = 1$, *then* every *computation of* $\Pi(w)$ *is an accepting computation.*

The family $\mathbf{\Pi}$ is said to provide a (polynomial–time) *uniform solution* to the problem X.

As a direct consequence of working with recognizer membrane systems, these complexity classes are closed under complement. Moreover, they are closed under polynomial time reductions [7].

The proof of the Milano theorem [9] (*each deterministic P system with active membranes but without membrane division can be simulated by a deterministic Turing machine with a polynomial slowdown*) and a proof given by A.E. Porreca [8] (*each tractable problem can be solved in polynomial time by families of recognizer P systems with active membranes and without input*) can be adapted in order to show the following result:

Corollary 3. $\mathbf{PMC}_{\mathcal{NAM}^0} = \mathbf{P}$.

This result suggests the following question: whether in the framework of polarizationless P systems with active membranes, division rules (only for elementary membranes) provide a borderline between tractability and intractability (assuming that **P** $\neq$ **NP**).

Let us recall that $\mathbf{P} \subseteq \mathbf{PMC}_{\mathcal{T}}$, where $\mathcal{T}$ is the class of all *basic transition* P systems (that is, transition P systems with only evolution, communication, and dissolution rules). In this kind of P systems the size of the membrane structure does not increase. The proof of this result can also be adapted to prove $\mathbf{PMC}_{\mathcal{NAM}} = \mathbf{P}$. It is well known that computationally hard problems can be solved in a uniform way by families of P systems from $\mathcal{AM}$. Therefore, in the framework of P systems with active membranes and three electrical charges, division rules (only for elementary membranes) provide a borderline between tractability and intractability (assuming that **P** $\neq$ **NP**).

2. A Conjecture of Păun

At the beginning of 2005, Gh. Păun (problem **F** from [5]) wrote:

> *My favorite question (related to complexity aspects in P systems with active membranes and with electrical charges) is that about the number of polarizations. Can the polarizations be completely avoided? The feeling is that this is not possible – and such a result would be rather sound: passing from no polarization to two polarizations amounts to passing from non–efficiency to efficiency.*

This so–called Păun's conjecture can be formally formulated in terms of membrane computing complexity classes as follows: $\mathbf{P} = \mathbf{PMC}_{\mathcal{AM}^0}$.

An affirmative answer to the conjecture would indicate that the ability to create an exponential amount of workspace (expressed in terms of the number of membranes and objects) in polynomial time, is not enough in order to solve computationally hard problems efficiently.

Conversely, a negative answer to the conjecture would provide a borderline between tractability and intractability (assuming that **P** $\neq$ **NP**): division rules for elementary membranes.

In this paper we provide a partial affirmative answer of the Păun's conjecture in the case that dissolution rules are forbidden. The proof of this result is based on the concept of dependency graph, improving and formalizing the result presented in [1].

Let Π be a recognizer polarizationless P system with active membranes which do not make use of dissolution rules. A directed graph can be associated with Π verifying the following property: every accepting computation of Π is characterized by the existence of a path in the graph between two specific nodes.

Definition 4. *Let* $\Pi = (\Gamma, H, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, R_\Pi, env)$*, be a recognizer P system with active membranes without polarizations and without dissolution. The dependency graph associated with* Π *is the directed graph* $G_\Pi = (V_\Pi, E_\Pi)$ *defined as follows:*

$V_\Pi = VL_\Pi \cup VR_\Pi \cup \{s_\Pi\}$, $s_\Pi \notin \Gamma \cup H$,

$$\begin{aligned} VL_\Pi = \{(a,h) \in \Gamma \times H :\ & \exists u \in \Gamma^* \ ([a \to u]_h \in R_\Pi) \vee \\ & \exists b \in \Gamma \ ([a]_h \to [\,]_h b \in R_\Pi) \vee \\ & \exists b \in \Gamma \ \exists h' \in ch(h) \ (a[\,]_{h'} \to [b]_{h'} \in R_\Pi) \vee \\ & \exists b, c \in \Gamma \ ([a]_h \to [b]_h[c]_h \in R_\Pi))\}, \\ VR_\Pi = \{(b,h) \in \Gamma \times H :\ & \exists a \in \Gamma \ \exists u \in \Gamma^* \ ([a \to u]_h \in R_\Pi \wedge b \in \mathit{alph}(u)) \vee \\ & \exists a \in \Gamma \ \exists h' \in ch(h) \ ([a]_{h'} \to [\,]_{h'} b \in R_\Pi) \vee \\ & \exists a \in \Gamma \ (a[\,]_h \to [b]_h \in R_\Pi) \vee \\ & \exists a, c \in \Gamma \ ([a]_h \to [b]_h[c]_h \in R_\Pi \vee [a]_h \to [c]_h[b]_h \in R_\Pi)\}, \end{aligned}$$

$E_\Pi = E^1_\Pi \cup E^2_\Pi$,

$E^1_\Pi = \{(s_\Pi, (a,h)) :$ *the object* a *is in membrane* h *at the initial configuration of* $\Pi\}$,

$$\begin{aligned} E^2_\Pi = \{((a,h),(b,h')) : & \ \exists u \in \Gamma^* \ ([a \to u]_h \in R_\Pi \wedge b \in \mathit{alph}(u) \wedge h = h') \vee \\ & ([a]_h \to [\,]_h b \in R_\Pi \ \wedge \ h' = f(h)) \vee \\ & (a[\,]_{h'} \to [b]_{h'} \in R_\Pi \ \wedge h \ = f(h')) \vee \\ & \exists c \in \Gamma \ (([a]_h \to [b]_h[c]_h \in R_\Pi \vee [a]_h \to [c]_h[b]_h \in R_\Pi) \wedge h = h')\}. \end{aligned}$$

This definition modifies the Definition 10 from [1] slightly.

Bearing in mind that in recognizer P systems all computations halt, we deduce that each path in the dependency graph associated with it must be simple.

Proposition 5. *There exists a deterministic Turing machine working in polynomial time that construct the dependency graph associated with each recognizer P system with active membranes without polarizations and without dissolution.*

Proof. A deterministic algorithm that given a recognizer P system constructs the associated dependency graph, is the following:

```
Input: $\Pi = (\Gamma, H, \mu, \mathcal{M}_1, \dots, \mathcal{M}_q, R_\Pi, env) \in \mathcal{AM}^0(-d)$
$V_\Pi \leftarrow \{s_\Pi\}$;  $E_\Pi \leftarrow \emptyset$
for $i \leftarrow 1$ to $q$ do
    for each $a \in supp(\mathcal{M}_i)$ do
        $V_\Pi \quad V_\Pi \cup \{(a, l(i))\}$;  $E_\Pi \quad E_\Pi \cup \{(s_\Pi, (a, l(i)))\}$
for each rule $r \in R_\Pi$ do
  if $r = [a \to u]_h \wedge \texttt{alph}(u) = \{a_1, \dots, a_s\}$ then
    $V_\Pi \quad V_\Pi \cup \bigcup_{j=1}^{s} \{(a, h), (a_j, h)\}$;  $E_\Pi \quad E_\Pi \cup \bigcup_{j=1}^{s} \{((a, h), (a_j, h))\}$
  if $r = [a]_h \to [\ \ ]_h b$ then
    $V_\Pi \quad V_\Pi \cup \{(a, h), (b, f(h))\}$;
    $E_\Pi \leftarrow E_\Pi \cup \{((a, h), (b, f(h)))\}$
  if $r = a[\ \ ]_h \to [b]_h$ then
    $V_\Pi \quad V_\Pi \cup \{(a, f(h)), (b, h)\}$;
    $E_\Pi \leftarrow E_\Pi \cup \{((a, f(h)), (b, h))\}$
  if $r = [a]_h \to [b]_h [c]_h$ then
    $V_\Pi \quad V_\Pi \cup \{(a, h)), (b, h), (c, h)\}$;
    $E_\Pi \leftarrow E_\Pi \cup \{((a, h)), (b, h)), ((a, h), (c, h))\}$
```

The running time of this algorithm is of the order $O(q \cdot |R_\Pi| + m \cdot |R_\Pi|)$, where q is the degree of the P system Π, and m is the value $\max\{length(r) : r \in R_\Pi\}$. As $|R_\Pi| \leq 2q \cdot |\Gamma|$, the total cost time of the algorithm is $O((q + m) \cdot q \cdot |\Gamma|)$. □

Next, let us prove that the dependency graph associated with a P system from $\mathcal{AM}^0(-d)$ can be used to characterize the behavior of the system (accepting or rejecting inputs), through the analysis of simple paths.

Theorem 6. *Let* $\Pi = (\Gamma, H, \mu, \mathcal{M}_1, \dots, \mathcal{M}_q, R_\Pi, env)$ *be a recognizer P system from* $\mathcal{AM}^0(-d)$*. The following assertions are equivalent:*

(1) There exists an accepting computation of Π.

(2) There exists a simple path in the dependency graph G_Π from s_Π to (yes, env) with length greater or equal to 2.

Proof. In order to prove that $(1) \implies (2)$ we will see that for each $n \geq 1$ and for each computation $\mathcal{C} = (C_0, C_1, \ldots, C_n)$ of Π, if $\mathcal{C}$ is an accepting computation, then there exists a simple path in the dependency graph G_Π from s_Π to (yes, env).

By induction on i. If $\mathcal{C} = (C_0, C_1)$ is an accepting computation of Π, then at the only step of that computation, the object yes has been sent to in the environment by a rule of the form $[a]_{h_s} \to yes\ [\ ]_{h_s}$, where h_s is the label of the skin membrane. Of course, in order to apply the rule, object a must occur in the skin in the initial configuration. Thus $((s_\Pi, (a, h_s)), ((a, h_s), (yes, env))$ is a simple path in G_Π from s_Π to (yes, env) with length greater or equal than 2. That is, the result holds for $n = 1$.

Let $n \geq 1$ and let us suppose that the result holds for n. Let $\mathcal{C} = (C_0, C_1, \ldots, C_n, C_{n+1})$ be an accepting computation of Π. Let us consider the recognizer P system $\Pi' = (\Gamma, H, \mu, \mathcal{M}'_1, \ldots, \mathcal{M}'_q, R_\Pi, env)$ where for each i, $1 \leq i \leq q$, $\mathcal{M}'_i$ is the contents of region i in the configuration C_1. Then $\mathcal{C}' = (C_1, \ldots, C_n, C_{n+1})$ is an accepting computation of Π′ of length n. By the induction hypothesis, there exists a simple path γ' in the dependency graph $G_{\Pi'}$ from $s_{\Pi'}$ to (yes, env) with length greater or equal than 2. Let (a_1, h_1) the next vertex of $s_{\Pi'}$ in that path. Then, the object a_1 belongs to the region i labeled by h_1 in the configuration C_1. We consider two cases:

– The object a_1 is in region labeled by h_1 in the configuration C_0.

  In this case, substituting the vertex $s_{\Pi'}$ by s_Π in γ' we obtain a simple path in the dependency graph G_Π from s_Π to (yes, env) with length greater or equal than 2.

– The object a_1 is not in region labeled by h_1 in the configuration C_0.

  In this case, the pair (a_1, h_1) has been produced at the first step of the computation $\mathcal{C}$, that is, there exist $(a_0, h_0) \in \Gamma \times H$ and a rule $r \in R_\Pi$ such that by applying r the pair (a_0, h_0) produces (a_1, h_1). Hence,

$$((s_\Pi, (a_0, h_0)), ((a_0, h_0), (a_1, h_1)), \gamma' - \{(s_{\Pi'}, (a_1, h_1))\})$$

  is a simple path in the dependency graph G_Π from s_Π to (yes, env) with length greater or equal than 2.

In order to prove that $(2) \implies (1)$ we will see that for each $n \geq 2$ and for each simple path γ in the dependency graph G_Π from s_Π to (yes, env) with length n, there exists an accepting computation of Π.

Let us reason again by induction on i. In the base case, let γ be a simple path in G_Π from s_Π to (yes, env) with length 2. Then, there exists an object a_0 in a region labeled by h_0 in the initial configuration such that $\gamma = ((s_\Pi, (a_0, h_0)), ((a_0, h_0), (yes, env)))$. Let C_1 a configuration obtained from C_0 by applying a rule of Π that yields (yes, env) from (a_0, h_0) (of course, many other rules could be applied at the same step in a maximally parallel way). Then, $\mathcal{C} = (C_0, C_1)$ is an accepting computation of Π.

Let $n \geq 2$ and let us suppose that the result holds for n. Let

$$\gamma = ((s_\Pi, (a_0, h_0)), ((a_0, h_0), (a_1, h_1)), \ldots, ((a_{n-1}, h_{n-1}), (yes, env)))$$

be a simple path in G_Π from s_Π to (yes, env) with length $n+1$.

For each j, $j = 0, 1, \ldots, n-1$, we denote by r_j a rule of Π transforming (a_j, h_j) into (a_{j+1}, h_{j+1}), where $a_n = yes$ and $h_n = env$. Let C_1 a configuration obtained from the initial configuration C_0 from Π by applying a multiset of rules containing r_0. Let Π' be the recognizer P system:

$$\Pi' = (\Gamma, H, \mu, \mathcal{M}'_1, \ldots, \mathcal{M}'_q, R_\Pi, env)$$

where for each i, $1 \leq i \leq q$, $\mathcal{M}'_i$ is the contents of region i in the configuration C_1. Then,

$$\gamma' = ((s_{\Pi'}, (a_1, h_1)), ((a_1, h_1), (a_2, h_2)), \ldots, ((a_{n-1}, h_{n-1}), (yes, env)))$$

is a simple path in $G_{\Pi'}$ from $s_{\Pi'}$ to (yes, env) with length n. From the induction hypothesis we deduce that there exists an accepting computation $\mathcal{C}' = (C_1, C_2, \ldots, C_t)$ of Π'. Therefore, $\mathcal{C} = (C_0, C_1, C_2, \ldots, C_t)$ is an accepting computation of Π. □

Corollary 7. *Let $X = (I_X, \theta_X)$ be a decision problem. Let $\mathbf{\Pi} = \{\Pi(n) : n \in \mathbf{N}\}$ be a family of recognizer P systems from $\mathcal{AM}^0(-d)$ solving X in a uniform way and in polynomial time. Let (cod, s) be a polynomial encoding associated with that solution. Then, for each instance $w \in I_X$ the following assertions are equivalent:*

(1) $\theta_X(w) = 1$.
(2) There exists a simple path in the dependency graph associated with $\Pi' = \Pi(s(w)) + cod(w)$, from $s_{\Pi'}$ to (yes, env).

Proof. It is enough to take into account the previous theorem and to notice that $\theta_X(w) = 1$ if and only if there exists an accepting computation in the system $\Pi(s(w))$ with input multiset $cod(w)$. □

Theorem 8. $\mathbf{PMC}_{\mathcal{AM}^0(-d)} = \mathbf{P}$.

Proof. Let us notice that $\mathbf{P} \subseteq \mathbf{PMC}_{\mathcal{AM}^0(-d)}$ because the class $\mathbf{PMC}_{\mathcal{AM}^0(-d)}$ is closed under polynomial–time reduction and contains the **P**–complete problem `CIRCUIT VALUE`.

Let us see the converse inclusion. Let $X \in \mathbf{PMC}_{\mathcal{AM}^0(-d)}$. Let $\mathbf{\Pi} = \{\Pi(n) : n \in \mathbf{N}\}$ be a family of recognizer P systems from $\mathcal{AM}^0(-d)$ solving X in a uniform way and in polynomial time. Let (cod, s) be a polynomial encoding associated with that solution. Let us see that $X \leq^p$ `REACHABILITY`.

Let w be an instance of X. Let us consider the system $\Pi(s(w))$ with input multiset $cod(w)$. Then we define:

$$F(w) = (G_{\Pi(s(w))+cod(w)}, s_{\Pi(s(w))+cod(w)}, (yes, env))$$

From Proposition 5 we deduce that F is a polynomial–time computable function. Moreover, we have:

$$w \in L_X \Longleftrightarrow \theta_X(w) = 1 \overset{Cor. 7}{\Longleftrightarrow} F(w) \in L_{\texttt{REACHABILITY}}$$

Finally, we deduce that $X \in \mathbf{P}$ because of the class **P** is closed under polynomial–time reductions, $X \leq^p$ `REACHABILITY` and `REACHABILITY` $\in \mathbf{P}$. □

Therefore, the feeeling expressed by Gh. Păun on his conjecture could be reformulated as follows: in the framework on polarizationless P systems with active membranes, passing from allowing dissolution rules to forbidding them amounts to passing from non-efficiency to efficiency (assuming that **P** $\neq$ **NP**).

3. Conclusions

Dissolution rules can be avoided when we solve **NP**–complete problems in an uniform way and in polynomial time by families of P systems with active membranes and three electrical charges [6].

This paper is partially answering a conjecture formulated by Gheorghe Păun at the beginning of 2005 related with the role played by the polarizations in the framework of P systems with active membranes from a computational complexity point if view. We present a detailed proof of a partial affirmative answer that highlights the relevance (in $\mathcal{AM}^0$) of an apparently innocent ingredient: dissolution rules.

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.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, A. Riscos-Núñez, F.J. Romero-Campero. On the power of dissolution in P systems with active membranes. In R. Freund, Gh. Paun, Gr. Rozenberg, A. Salomaa (eds.) *Membrane Computing, 6th International Workshop, WMC 2005, Vienna, Austria, July 18-21, 2005, Revised Selected and Invited Papers*. Lecture Notes in Computer Science, **3850** (2006), 224-240.

[2] Ch. H. Papadimitriou. *Computational Complexity*, Addison-Wesley, California, USA, 1995.

[3] Gh. 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.

[4] Gh. Păun. P systems with active membranes: Attacking **NP**–complete problems. *Journal of Automata, Languages and Combinatorics*, **6**, 1 (2001), 75–90.

[5] Gh. Păun. Further twenty six open problems in membrane computing. In M.A. Gutiérrez, A. Riscos, F.J. Romero, D. Sburlan (eds.) *Proceedings of the Third Brainstorming Week on Membrane Computing*, Report RGNC 01/04, Fénix Editora, Sevilla, 2005, pp. 249–262.

[6] M.J. Pérez–Jiménez, A. Romero–Jiménez, F. Sancho–Caparrini. Complexity classes in cellular computing with membranes. *Natural Computing*, **2**, 3 (2003), 265–285.

[7] M.J. Pérez-Jiménez, A. Romero-Jiménez, F. Sancho-Caparrini. A polynomial complexity class in P systems using membrane division. *Journal of Automata, Languages and Combinatorics*, **11**, 4 (2006), 423-434. A preliminary version in E. Csuhaj-Varjú, C. Kintala, D. Wotschke, Gy. Vaszil (eds.) *Proceedings of the Fifth International Workshop on Descriptional Complexity of Formal Systems, DCFS 2003*, Budapest, Hungary, July 12-14, 2003, pp. 284-294.

[8] A. E. Porreca. *Computational Complexity Classes for Membrane Systems*, Master Degree Thesis, Universita' di Milano-Bicocca, Italy, 2008.

[9] C. Zandron, C. Ferretti, G. Mauri. Solving NP–complete problems using P systems with active membranes. In I. Antoniou, C.S. Calude, M.J. Dinneen (eds.) *Unconventional Models of Computation*, Springer, Berlin, 2000, pp. 289–301.