Complexity Classes in Cellular Computing with Membranes

M. J. Pérez-Jiménez, A. Romero Jiménez and F. Sancho-Caparrini

Dpto. Ciencias de la Computación e Inteligencia Artificial
E.T.S. Ingeniería Informática. Universidad de Sevilla
Avda. Reina Mercedes s/n. 41012 - Sevilla. España

{Mario.Perez,Alvaro.Romero,Fernando.Sancho}@cs.us.es

Abstract. In this paper we introduce the complexity class $\mathbf{PMC}^*_{\mathcal{F}}$ of all decision problems solvable in polynomial time by a family of P systems belonging to a prefixed class of recognizer membrane systems, $\mathcal{F}$.

1 Introduction

One of the main goals of *Complexity Theory* is to provide tools allowing to classify the problem regarding to the amount of resources needed for their resolution. In this paper the first *complexity classes* for P systems are presented, inspired in some ideas that Gh. Păun [1] discussed with the authors. Theses classes allows us to detect some of the inherent difficulties to the computational resolution os certain problems, and they provide a classification of the *abstract problems* according the resources they need to be solved in a given model of Cellular Computing with Membranes. Of course, such a classification demands a precise and formal definition about the concept of *abstract problem* and the model to be considered.

The following parameters used to specify a complexity class:

- The *model* of computation (in our case, recognizer P systems).
- The *mode* of computation (in our case, deterministic and parallel).
- The *measure* of complexity (time and space).
- An upper *bound* of the resources (a total recursive function from $\mathbf{N}^+$ to $\mathbf{N}^+$).

The paper is organized as follows: in the next section we define what a decision problem is as well as we understand by a recognizer P system without or with input which will the models considered to study the complexity classes. In section 3 the complexity classes associated with P system without input is introduced. Section 4 is devoted to study the complexity classes associated with P systems with input.

2 Preliminaries

Definition 1. *A decision problem, X, is a pair (I_X, θ_X) such that I_X is a language over a finite alphabet (whose elements are called instances) and θ_X is a total boolean function over I_X.*

Before to give the precise definition of the classes, we must to present some previous notions about P systems.

Definition 2. *A* recognizer P system *is a pair* $\big(\Pi, \theta_\Pi\big)$ *such that* Π *is a P systems and* θ_Π *is a total boolean function over the halting computations of* Π*. Let* $\mathcal{C}$ *be a halting computation of* Π*. If* $\theta_\Pi(\mathcal{C}) = 1$ *we will say that* $\mathcal{C}$ *is an accepting computation. If* $\theta_\Pi(\mathcal{C}) = 0$*, we will say that* $\mathcal{C}$ *is a rejecting computation.*

Usually, we will identify the recognizer P system $\big(\Pi, \theta_\Pi\big)$ with the P system Π, because the function θ_Π is usually given in an implicit way by means of the semantic of Π.

Definition 3. *Let* $\mathbf{\Pi} = \big(\Pi(i)\big)_{i \in I}$ *a family of P systems. Then,*

1. *We will say that* $\mathbf{\Pi}$ *is* $\mathcal{D}$-consistent *if for each* $i \in I$*,* $\Pi(i) \in \mathcal{D}$*.*
2. *We will say that* $\mathbf{\Pi}$ *is* uniform*, by Turing machines, if there exists such a machine that, for a given* $i \in I$*, it constructs deterministically the P system* $\Pi(i)$*. If this construction is made in polynomial time in the size of* i*, we will say that a* $\mathbf{\Pi}$ *is* polynomially uniform*, by Turing machines.*

Moreover, we will demand to the P systems in this paper to be confluent, in the following sense.

Definition 4. *We say that a recognizer P system,* $\big(\Pi, \theta_\Pi\big)$*, is* confluent*, if it verifies the following conditions:*

1. *Every computation of* Π *is a halting computation.*
2. *If* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *are two computations of* Π *with the same initial configuration, then* $\theta_\Pi(\mathcal{C}_1) = \theta_\Pi(\mathcal{C}_2)$*.*

That is, a recognizer P system is confluent if, either all its computations with identical initial configurations are accepting computations, or all of them are rejecting ones. Note that a deterministic recognizer P system is confluent if and only if every computation is a halting computation.

Definition 5. *A P system with input is a tuple* (Π, Σ, i_Π)*, where:*

- Π *is a P system, with working alphabet* Γ *and initial multisets* $\mathcal{M}_1, \ldots, \mathcal{M}_p$ *(associated with membranes labeled by* $1, \ldots, p$*, respectively).*
- Σ *is an (input) alphabet strictly contained in* Γ*.*
- $\mathcal{M}_1, \ldots, \mathcal{M}_p$ *are multisets over* $\Gamma - \Sigma$*.*
- i_Π *is the label of a distinguished membrane (of input).*

Definition 6. *Let* (Π, Σ, i_Π) *be a P system with input and without external output. Let* Γ *be the working alphabet of* Π*,* μ *the membrane structure and* $\mathcal{M}_1, \ldots, \mathcal{M}_p$ *the initial multisets (over* $\Gamma - \Sigma$*) of* Π*. Let* $m \in M(\Sigma)$ *be a multiset over* Σ*. The initial configuration of* (Π, Σ, i_Π) *with input* m *is* (μ_0, M_0)*, where* $\mu_0 = \mu$*,* $M_0(j) = \mathcal{M}_j$*, for each* $j \neq i_\Pi$*, and* $M_0(i_\Pi) = \mathcal{M}_{i_\Pi} \cup m$*.*

Note 1. Let us denote by I_Π the set of all inputs of the P system Π. That is, $I_\Pi = M(\Sigma)$, the collection of the multisets over Σ.

The computations of a P system with input $m \in M(\Sigma)$ are defined in a natural way. The only novelty is that the initial configuration must be the initial configuration of the system associated with the input multiset $m \in M(\Sigma)$.

In the case of P systems with input and with external output (where we can imagine that the internal processes are unknown, and we only obtain the information that the system sends out to the environment), the concept of computation is introduced in a similar way but with a slightly variant. In the configurations, we will not work directly with the membrane structure μ but with another structure associated with it including, in some sense, the environment.

Definition 7. *Let $\mu = (V(\mu), E(\mu))$ be a membrane structure. The* membrane structure with external environment *associated with μ is the rooted tree $Ext(\mu)$ such that: (a) the root of the tree is a new node that we will denote env; (b) the set of nodes is $V(\mu) \cup \{env\}$; and (c) the set of edges is $E(\mu) \cup \{\{env, skin\}\}$. The node env is called* external environment *of the structure μ.*

Note that we have only included a new node representing the environment which is only connected with the skin, while the original membrane structure remains unchanged. In this way, every configuration of the system informs about the environment and its content.

Definition 8. *A recognizer P system with input is a P system with input, (Π, Σ, i_Π), and with external output such that:*

1. *The working alphabet contains two distinguished elements Yes, No.*
2. *All of its computations halt.*
3. *If $\mathcal{C}$ is a computation of Π, then either some object Yes or some object No (but not both) must have been sent out to the external environment, and only in the last step of the computation. We say that $\mathcal{C}$ is an accepting computation (respectively, rejecting computation) if the object Yes (respectively, No) appears in the external environment associated with the corresponding halting configuration of $\mathcal{C}$.*

Definition 9. *We say that a recognizer P system with input is* confluent*, if for every $\mathcal{C}_1$, $\mathcal{C}_2$ computations of Π with the same initial configuration with input m, it is verified that $\theta_\Pi(\mathcal{C}_1) = \theta_\Pi(\mathcal{C}_2)$.*

3 Complexity classes associated with P systems without input

The first results about *resolution* of **NP**-complete problems in polynomial time (even linear time) by using cellular computing devices have been obtained by means of variants of P systems without input membrane. Because of this, in the proofs of these results what is constructed is, in fact, for each instance of the problem one system solving it. From this idea, we present the following definition.

Definition 10. *Let $\mathcal{D}$ be a class of recognizer P systems without input, $g : \mathbf{N}^+ \to \mathbf{N}^+$ a total recursive function, and $X = (I_X, \theta_X)$ a decision problem. We say that $X \in \mathbf{MC}^{\mathbf{f}}_{\mathcal{D}}(g)$ if there exists a family, $\mathbf{\Pi} = (\Pi(w))_{w \in I_X}$, of P systems such that:*

- *$\mathbf{\Pi}$ is $\mathcal{D}$–consistent.*
- *$\mathbf{\Pi}$ is polynomially uniform by Turing machines.*
- *The family $\mathbf{\Pi}$ is (X, g)-bounded; that is, for each $w \in I_X$, all computations of $\Pi(w)$ halt in, at most, $g(|w|)$ steps.*
- *The family $\mathbf{\Pi}$ is X-sounded; that is, for each $w \in I_X$, if $\theta_X(w) = 1$, then every computation of $\Pi(w)$ is an accepting computation.*
- *The family $\mathbf{\Pi}$ is X-complete; that is, for each $w \in I_X$, if there exists an accepting computation of $\Pi(w)$, then $\theta_X(w) = 1$.*

If we consider in the above definition the collection of polynomial functions as bounded functions, then we obtain the polynomial complexity class for families of P systems without input.

Definition 11. *The class of decision problems solvable in polynomial time by a family of P systems without input is the class*

$$\mathbf{PMC}^f_{\mathcal{D}} = \bigcup_{k>0} \mathbf{MC}^f_{\mathcal{D}}(n^k)$$

It must be noted that this class is closed under polynomial time reduction.

Proposition 1. *Let $X = (I_X, \theta_X)$ and $Y = (I_Y, \theta_Y)$ two decision problems such that X is polynomial time reducible to Y. If $Y \in \mathbf{PMC}^f_{\mathcal{D}}$ then $X \in \mathbf{PMC}^f_{\mathcal{D}}$.*

Proof. Since X is polynomial time reducible to Y, there exists a polynomial computable function $h : I_X \to I_Y$, such that for every $w \in I_X$, $\theta_X(w) = 1$ if and only if $\theta_Y(h(w)) = 1$. Hence, there exists a natural number $k_h > 0$ such that $|h(w)| \leq |w|^{k_h}$, for every $w \in I_X$.

Since, by hypothesis, $Y \in \mathbf{PMC}^f_{\mathcal{D}}$, there exist a natural number $k > 0$ and a family of P systems, $\mathbf{\Pi}_Y = \left(\Pi_Y(w')\right)_{w' \in I_Y}$ such that:

1. The family $\mathbf{\Pi}_Y$ is $\mathcal{D}$-consistent.
2. For each $w' \in I_Y$, the system $\Pi(w')$ has no input membrane.
3. The family $\mathbf{\Pi}_Y$ is polynomially uniform.
4. The family $\mathbf{\Pi}_Y$ is (Y, n^k)-bounded.
5. The family $\mathbf{\Pi}_Y$ is Y-sounded.
6. The family $\mathbf{\Pi}_Y$ is Y-complete.

Let us consider the family $\mathbf{\Pi} = \left(\Pi(w)\right)_{w \in I_X}$, where $\Pi(w) = \Pi_Y(h(w))$, for every $w \in I_X$. Then it is verified that:

1. The family $\mathbf{\Pi}$ is polynomially uniform: it is enough to have in mind that the family $\mathbf{\Pi}_Y$ is polynomially uniform and the function h is polynomial time computable.

2. The family $\mathbf{\Pi}$ is $(X, n^{k_h \cdot k})$-bounded: let $w \in I_X$, since $\Pi(w) = \Pi_Y(h(w))$, every computation of $\Pi(w)$ is a halting computation, and it works in less than $|h(w)|^k \leq (|w|^{k_h})^k = |w|^{k_h \cdot k}$ steps.
3. The family $\mathbf{\Pi}$ is X-sounded: let $w \in I_X$ such that $\theta_X(w) = 1$; then $\theta_Y(h(w)) = 1$, hence every computation of $\Pi_Y(h(w)) = \Pi(w)$ is an accepting one.
4. The family $\mathbf{\Pi}$ is X-complete: let $w \in I_X$ such that there exists an accepting computation of $\Pi(w) = \Pi_Y(h(w))$. Then $\theta_Y(h(w)) = 1$, and hence $\theta_X(w) = 1$.

Consequently, $X \in \mathbf{PMC}^f_{\mathcal{D}}$.

$\square$

4 Complexity classes associated with P systems with input

In the above definition of complexity class we had to work with a family of recognizer P systems without input, because each P system can only decide about one instance of the problem.

If we consider P systems with input membrane, it is possible to define a complexity class in some similar way to that defined for classic models (i.e. Turing Machines).

Definition 12. *Let $\mathcal{D}$ be a class of recognizer P systems with input, $g : \mathbf{N}^+ \to \mathbf{N}^+$ a total recursive function, and $X = (I_X, \theta_X)$ a decision problem. We say that $X \in \mathbf{MC}^{\mathbf{i}}_{\mathcal{D}}(g)$ if there exists a P system, Π such that:*

- *$\Pi \in \mathcal{D}$.*
- *Π can be constructed in polynomially time by a deterministic Turing machine, depending on the problem.*
- *There exists a polynomially computable function, $cod : I_X \to I_\Pi$, such that:*
  - *Π is (X, g)-bounded regarding to cod; that is, for each $w \in I_X$, every computation of Π with input $cod(w)$ is a halting computation, and it halts in less than $g(|w|)$ steps.*
  - *Π is X-sounded regarding to cod; that is, for each $w \in I_X$, if $\theta_X(w) = 1$ then every computation of Π with input $cod(w)$ is an accepting computation.*
  - *Π is X-complete regarding to cod; that is, for each $w \in I_X$, if there exists an accepting computation of Π with input $cod(w)$ then $\theta_X(w) = 1$.*

If we consider as bound functions the polynomial functions, we obtain the polynomial complexity class for P systems with input.

Definition 13. *The class of decision problems solved in polynomial time for a P system with input is the class*

$$\mathbf{PMC}^i_{\mathcal{D}} = \bigcup_{k>0} \mathbf{MC}^i_{\mathcal{D}}(n^k)$$

This class is, also, closed under polynomial time reduction.

Proposition 2. *Let $X = (I_X, \theta_X)$ and $Y = (I_Y, \theta_Y)$ two decision problems such that X is reducible to Y in polynomial time. If $Y \in \mathbf{PMC}^i_{\mathcal{D}}$ then $X \in \mathbf{PMC}^i_{\mathcal{D}}$.*

Proof. Since X is reducible to Y in polynomial time, there exists a computable function $h : I_X \to I_Y$ in polynomial time, such that for every $w \in I_X$, $\theta_X(w) = 1$ if and only if $\theta_Y(h(w)) = 1$. Hence, there exists a natural number $k_h > 0$ such that $|h(w)| \leq |w|^{k_h}$, for every $w \in I_X$.

Since, by hypothesis, $Y \in \mathbf{PMC}^i_{\mathcal{D}}$, there exist a natural number $k > 0$ and a P system, Π, such that:

1. $\Pi \in \mathcal{D}$.
2. Π can be constructed in polynomial time by a Turing Machine.
3. There exists a polynomially computable function, $cod_Y : I_Y \to I_\Pi$, such that:
   - Π is (Y, n^k)-bounded regarding to cod_Y.
   - Π is Y-sounded regarding to cod_Y.
   - Π is Y-complete regarding to cod_Y.

Let us consider the function $cod = cod_Y \circ h : I_X \to I_\Pi$. Then cod is a computable function in polynomial time. Moreover, the following conditions are verified:

- Π is $(X, n^{k_h \cdot k})$-bounded regarding to *cod*: let $w \in I_X$; then every computation of Π with input $cod(w) = cod_Y(h(w))$ halts in, at most, $|h(w)|^k \leq \left(|w|^{k_h}\right)^k = |w|^{k_h \cdot k}$ steps.
- Π is X-sounded regarding to *cod*: let $w \in I_X$ such that $\theta_X(w) = 1$; then $\theta_Y(h(w)) = 1$, hence every computation of Π with input $cod_Y(h(w)) = cod(w)$ is a halting computation.
- Π is X-complete regarding to *cod*: let $w \in I_X$ such that there exists an accepting computation of Π with input $cod(w) = cod_Y(h(w))$, hence $\theta_Y(h(w)) = 1$, and therefore $\theta_X(w) = 1$.

Consequently, $X \in \mathbf{PMC}^i_{\mathcal{D}}$.

□

Although above definition of complexity class is rather natural, it is very usual in practice to find not one P system with input solving a decision problem, but a family of P systems with input, where one of its P systems solves the instances of *similar size*. This carried out the third complexity class included in this work.

Definition 14. *Let $X = (I_X, \theta_X)$ a decision problem, $\mathcal{D}$ a class of recognizer P systems with input and $\mathbf{\Pi} = (\Pi(n))_{n \in \mathbf{N}^+}$ a family of P systems of $\mathcal{D}$. A polynomial encoding of I_X in $\mathbf{\Pi}$ is a pair, (cod, s), of polynomial time computable functions, $cod : I_X \to \bigcup_{n \in \mathbf{N}^+} I_{\Pi(n)}$ and $s : I_X \to \mathbf{N}^+$, such that for every $w \in I_X$ we have $cod(w) \in I_{\Pi(s(w))}$.*

Definition 15. *Let $\mathcal{D}$ be a class of recognizer P systems with input, $g : \mathbf{N}^+ \to \mathbf{N}^+$ a total recursive function, and $X = (I_X, \theta_X)$ a decision problem. We say that $X \in \mathbf{MC}^{fi}_{\mathcal{D}}(g)$ if there exists a P system, $\mathbf{\Pi} = \big(\Pi(n)\big)_{n\in\mathbb{N}^+}$ such that:*

- *The family $\mathbf{\Pi}$ is $\mathcal{D}$-consistent.*
- *The family $\mathbf{\Pi}$ is polynomially uniform.*
- *There exist a polynomial encoding, (cod, s), of I_X in $\mathbf{\Pi}$, such that*
  - *The family $\mathbf{\Pi}$ is (X, g)-bounded regarding to (cod, s); that is, for each $w \in I_X$, every computation of $\Pi(s(w))$ with input $cod(w)$ halts in, at most, $s(|w|)$ steps.*
  - *The family $\mathbf{\Pi}$ is X-sounded regarding to (cod, s); that is, for every $w \in I_X$, if $\theta_X(w) = 1$ then every computation of $\Pi(s(w))$ with input $cod(w)$ is an accepting computation.*
  - *The family $\mathbf{\Pi}$ is X-complete regarding to (cod, s); that is, for every $w \in I_X$, if there exists an accepting computation of $\Pi(s(w))$ with input $cod(w)$ then $\theta_X(w) = 1$.*

As usual, the associated polynomial class can be obtained by using the polynomial functions as bound functions.

Definition 16. *The class of decision problems solved by a family of P systems with input is the class*

$$\mathbf{PMC}^{fi}_{\mathcal{D}} = \bigcup_{k>0} \mathbf{MC}^{fi}_{\mathcal{D}}(n^k)$$

This class is closed under polynomial time reduction.

Proposition 3. *Let $X = (I_X, \theta_X)$ and $Y = (I_Y, \theta_Y)$ two decision problems such that X is reducible to Y in polynomial time. If $Y \in \mathbf{PMC}^{fi}_{\mathcal{D}}$ then $X \in \mathbf{PMC}^{fi}_{\mathcal{D}}$.*

Proof. Since X is reducible to Y in polynomial time, there exists a total computable function $h : I_X \to I_Y$ in polynomial time, such that, for every $w \in I_X$, $\theta_X(w) = 1$ if and only if $\theta_Y(h(w)) = 1$. Hence, there exists a natural number $k_h > 0$ such that $|h(w)| \leq |w|^{k_h}$, for every $w \in I_X$.

Since, by hypothesis, $Y \in \mathbf{PMC}^{fi}_{\mathcal{D}}$, there exist a natural number $k > 0$ and a family of P systems, $\mathbf{\Pi} = \big(\Pi(n)\big)_{n\in\mathbb{N}^+}$, such that:

- The family $\mathbf{\Pi}$ is $\mathcal{D}$-consistent.
- The family $\mathbf{\Pi}$ is polynomially uniform.
- There exist a polynomial encoding, (cod_Y, s_Y), of I_Y in $\mathbf{\Pi}$.
- $\mathbf{\Pi}$ is (Y, n^k)-bounded regarding to (cod_Y, s_Y).
- $\mathbf{\Pi}$ is Y-sounded regarding to (cod_Y, s_Y).
- $\mathbf{\Pi}$ is Y-complete regarding to (cod_Y, s_Y).

Let us consider the functions $cod = cod_Y \circ h : I_X \to \bigcup_{n\in\mathbb{N}^+} I_{\Pi(n)}$ y $s = s_Y \circ h$. Then cod and s are computable in polynomial time. Moreover, the following conditions are verified:

- (cod, s) is a polynomial encoding of I_X in $\mathbf{\Pi}$: for each $w \in I_X$, $cod(w) = cod_Y(h(w)) \in I_{\Pi(s_Y(h(w)))} = I_{\Pi(s(w))}$.
- $\mathbf{\Pi}$ is $(X, n^{k_h \cdot k})$-bounded regarding to (cod, s): let $w \in I_X$, then every computation of $\Pi(s(w)) = \Pi(s_Y(h(w)))$ with input $cod(w) = cod_Y(h(w))$ halts in, at most, $|h(w)|^k \leq (|w|^{k_h})^k = |w|^{k_h \cdot k}$ steps.
- $\mathbf{\Pi}$ is X-sounded regarding to (cod, s): let $w \in I_X$ such that $\theta_X(w) = 1$, then $\theta_Y(h(w)) = 1$, hence every computation of $\Pi(s_Y(h(w))) = \Pi(s(w))$ with input $cod_Y(h(w)) = cod(w)$ is an accepting computation.
- $\mathbf{\Pi}$ is X-complete regarding to (cod, s): let $w \in I_X$ such that there exists an accepting computation of $\Pi(s(w)) = \Pi(s_Y(h(w)))$ with input $cod(w) = cod_Y(h(w))$, hence $\theta_Y(h(w)) = 1$, and therefore $\theta_X(w) = 1$.

Consequently, $X \in \mathbf{PMC}^{fi}_{\mathcal{D}}$.

□

It is easy to prove that the following relations between above classes are verified.

Proposition 4. *Let $\mathcal{D}$ be a class of recognizer P systems with input membrane and let $\mathcal{D}'$ be the same class but considered as without input membrane. It is verified:*

1. *If $X \in \mathbf{MC}^{fi}_{\mathcal{D}}(g)$ then $X \in \mathbf{MC}^{f}_{\mathcal{D}'}(g)$.*
2. *If $X \in \mathbf{MC}^{i}_{\mathcal{D}}(g)$ then $X \in \mathbf{MC}^{fi}_{\mathcal{D}}(g)$.*

Proof.

1. Let us suppose that $X \in \mathbf{MC}^{fi}_{\mathcal{D}}(g)$. Then, there exists a family of P systems, $\mathbf{\Pi} = (\Pi(n))_{n \in \mathbb{N}^+}$, verifying:
   - The family $\mathbf{\Pi}$ is $\mathcal{D}$-consistent.
   - $\mathbf{\Pi}$ is polynomially uniform.
   - There exist a polynomial encoding, (cod, s), of I_X in $\mathbf{\Pi}$.
   - $\mathbf{\Pi}$ is (X, g)-bounded regarding to (cod, s).
   - $\mathbf{\Pi}$ is X-sounded regarding to (cod, s).
   - $\mathbf{\Pi}$ is X-complete regarding to (cod, s).

   Let us consider the family $\mathbf{\Pi}' = (\Pi'(w))_{w \in I_X}$ in such way that, for each $w \in I_X$ the P system $\Pi'(w)$ is obtained considerending the P system $\Pi(s(w))$ as a P systems without input membrane, and with the multiset $cod(w)$ added to its input membrane.
   Then, it is easy to check that such a family verifies the conditions for $X \in \mathbf{MC}^{f}_{\mathcal{D}'}$.
2. Let us suppose that $X \in \mathbf{MC}^{i}_{\mathcal{D}}(g)$. Then there exist a P system, Π, verifying the following conditions:
   - $\Pi \in \mathcal{D}$.
   - Π can be constructed in polynomial time by a Turing machine.
   - There exists a computable function, $cod : I_X \to I_\Pi$, in polynomial time, such that:

- Π is (X, g)-bounded regarding to *cod*.
- Π is X-sounded regarding to *cod*.
- Π is X-complete regarding to *cod*.

Let us consider the family $\mathbf{\Pi} = \big(\Pi(n)\big)_{n\in\mathbb{N}^+}$, where for each $n \in \mathbb{N}^+$, $\Pi(n) = \Pi$. Then, it is easy to check that this family verifies the conditions for $X \in \mathbf{MC}^{fi}_{\mathcal{D}}$.

5 Conclusions

In this work several complexity classes associated with membrane systems (with or without input) have been presented. Besides, some inclusion relations between them have been studied.

We think that this paper establish a new framework to study computational complexity of decision problems through membrane systems, what can provide relevant information about the *effective computational power* of different variants of P systems.

Moreover, we think that these complexity classes in membrane systems offer a new way to attack the **P**$\neq$**NP** conjecture, now inside the framework of the cellular computing with membranes.

Acknowledgement

The authors wish to acknowledge the support of the project TIC2002-04220-C03-01 of the Ministerio de Ciencia y Tecnología of Spain, cofinanced by FEDER funds.

References

1. Păun, G.: *Membrane Computing. An introduction*, Springer-Verlag, Berlín, 2002.
2. Păun Gh. Computing with membranes, *Journal of Computer and System Sciences*, 61, 1 (2000), 108–143, and *Turku Center for Computer Science-TUCS Report* No 208, 1998 (www.tucs.fi).
3. Păun, Gh.; Rozenberg, G. A guide to membrane computing, *Theoretical Computer Science*, 287, 2002, 73–100.
4. Pérez–Jiménez, M.J.; Romero-Jiménez, A.; Sancho-Caparrini, F. *Teoría de la Complejidad en modelos de computación con membranas*, Ed. Kronos, Sevilla, I.S.B.N. 84-86273-57-9, 2002.