Graphics and P Systems: Experiments with JPLANT

Elena Rivero-Gil, Miguel A. Gutiérrez-Naranjo, Mario J. Pérez-Jiménez

Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla
Avda. Reina Mercedes s/n, 41012, Sevilla, Spain
E-mails: elen.rg@gmail.com, magutier@us.es, marper@us.es

Summary. The hand-made graphical representation of the configuration of a P system becomes a hard task when the number of membranes and objects increases. In this paper we present a new software tool, called JPLANT, for computing and representing the evolution of a P system model with membrane creation. We also present some experiments performed with JPLANT and point out new lines for the research in computer graphics with membrane systems.

1 Introduction

Since A.R. Smith [12] proposed the Lindenmayer systems (L-systems) [5] as a tool for synthesizing realistic images of plants, many efforts have been done for bridging the theory of formal languages and computer graphics.

In [2, 3], a first membrane-based device for computer graphics was presented. It was a hybrid model between L-systems and membrane computing and it used concepts very close to the L-systems model. Later, in [10], a new approach was presented for representing the development of higher plants with P systems. It was based on a type of P systems with membrane creation and it was entirely developed with membrane computing techniques. The basic idea was to consider the growing of the structure the membranes in a P system with membrane creation.

By definition, the structure of membranes in a cell-like P system is a tree. In P systems with membrane creation, new membranes can be created inside the existing membranes and this produces the expansion of the structure of membranes by increasing the depth of the branches. With an appropriate interpretation of the objects inside the membranes, the membrane structure can be represented as a tree which evolves in time and the length and width of the branches can grow in a similar way to real plants. In [11], the study started at [10] was completed by adding stochastic rules to the P system. In this case, the non-deterministic choice of different rules produces different configurations of the P systems and hence, different graphical representations.

The hand-made graphical representation of the configuration of a P system becomes a hard task when the number of membranes and objects increases. For going on with the study of the relation between P systems and computer graphics it was necessary to develop a software able to deal with complex P systems and represent graphically its evolution in time.

In this paper we present such a software, JPLANT, which computes the first configurations of a computation and draws the corresponding graphical representation. This software is a very useful tool for the experimental research of the graphical representation of P systems. We show several experiments and open new research lines for exploring the possibilities of P systems.

The paper is organized as follows: Section 2 recall the restricted model of P systems with membrane creation used for the graphical design. Section 3 gives a brief presentation of the software JPLANT and the next section shows several experiments. The paper finishes with some conclusions and lines for future research.

2 P Systems with Membrane Creation

Membrane computing is a branch of natural computing which abstracts from the structure and the functioning of the living cell. In the basic model, membrane systems (also frequently called P systems) are distributed parallel computing devices, processing multisets of symbol-objects, synchronously, in the compartments defined by a cell-like membrane structure[1].

In this paper we will consider P systems which make use of membrane creation rules, which was first introduced in [4, 6]. However, our needs are far simpler than what the models found in the literature provide. This is the reason why we introduce the new variant of *restricted P systems with membrane creation.*

A restricted P system with membrane creation is a tuple $\Pi = (O, \mu, w_1, \dots, w_m, R)$ where:

1. O is the alphabet of *objects*. There exist two distinguished objects, F and W that always belong to the alphabet of any P system considered below.
2. μ is the initial *membrane structure*, consisting of a hierarchical structure of m membranes (all of them with the same label; for the sake of simplicity we omit the label).
3. $w_1, \dots, w_m$ are the multisets of objects initially placed in the m regions delimited by the membranes of μ.
4. R is a finite set of *evolution rules* associated with every membrane, which can be of the two following kinds:
   a) $a \to v$, where $a \in O$ and v is a multiset over O. This rule replaces an object a present in a membrane of μ by the multiset of objects v.

[1] A detailed description of P systems can be found in [9] and updated information in [13].

b) $a \to [v]$, where $a \in O$ and v is a multiset over O. This rule replaces an object a present in a membrane of μ by a new membrane with the same label and containing the multiset of objects v.

A membrane structure together with the objects contained in the regions defined by its membranes constitute a configuration of the system. A transition step is performed applying to a configuration the evolution rules of the system in the usual way within the framework of membrane computing, that is, in a non-deterministic maximally parallel way; a rule in a region is applied if and only if the object occurring in its left–hand side is available in that region; this object is then consumed and the objects indicated in the right–hand side of the rule are created inside the membrane. The rules are applied in all the membranes simultaneously, and all the objects in them that can trigger a rule must do it. When there are several possibilities to choose the evolution rules to apply, non-determinism takes place.

2.1 Graphical Representation

In this section we show how to use, through a suitable graphical representation, restricted P systems with membrane creation to model branching structures. The key point of the representation relies on the fact that a membrane structure is a *rooted tree of membranes*, whose root is the skin membrane and whose leaves are the elementary membranes. Thus, this seems a suitable frame to encode the branching structure.

Let us suppose that the alphabet O of objects contains the objects F and W, and let us fix the lengths l and w.

A simple model to graphically represent a membrane structure is to make a depth-first search of it, drawing, for each membrane containing the object F, a segment of length $m \times l$, where m is the multiplicity of F. If the number of copies of F in a membrane increases along the computation, the graphical interpretation is that the corresponding segment is lengthening. Analogously, the multiplicity of the symbol W specify the width of the segments to be drawn as follows: if the number of objects W present in a membrane is n, then the segment corresponding to this membrane must be drawn with width $n \times w$.

Each segment is drawn rotated with respect to the segment corresponding to its parent membrane. In order to determine the rotation angle we need to fix a third parameter δ. This angle δ together with the length l and the width w will determine the picture of the P system.

In order to compute the rotation angle of a segment with respect to its parent membrane we consider two new objects that can appear in the alphabet: $+$ and $-$. The rotation angle will be $n \times \delta$, where n is the multiplicity of objects "+" minus the multiplicity of objects "$-$" in the membrane. That is, each object "+" means that the rotation angle is increased by δ whereas each object "$-$" means that it is decreased by δ.

Inside the membranes other objects can appear that do not have geometrical interpretation. They are related to the development of the graph in time.

For a better understanding let us consider the following example: let Π_1 be the restricted P system with membrane creation such that

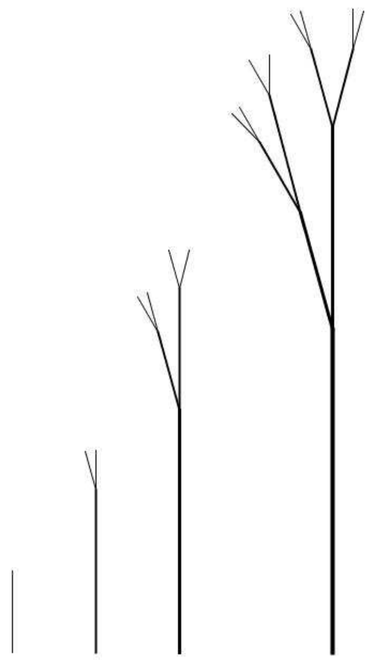

Fig. 1. First four configurations

- The alphabet of objects is $O = \{F, W, B_l, B_s, B_r, L, L_1, E, +, -\}$.
- The initial membrane structure together with the initial multiset of objects is $[F^2\, W\, B_l\, B_s\, L_1\, E]$.
- The rules are:

$$\begin{array}{llll} B_l & \rightarrow [+\, F\, W\, B_l\, B_s\, L\, E] & \quad L & \rightarrow L\, F \\ B_s & \rightarrow [F\, W\, B_l\, B_r\, L_1\, E] & \quad L_1 & \rightarrow L_1\, F^2 \\ B_r & \rightarrow [-\, F\, W\, B_l\, B_s\, L\, E] & \quad E & \rightarrow E\, W \end{array}$$

In this system, the object B_s represents the straight branches to be created, whereas the objects B_l and B_r represent branches to be created rotated to the left and to the right, respectively. The objects F and W will determine the length and the width of the corresponding branch. The objects L, L_1 and E do not have a graphical interpretation; they can be considered as seeds for growing the branch in length and width.

The initial configuration consists of one membrane which contains two copies of F and one copy of W. If we consider the parameters l, w and δ, then the graphical representation of this initial configuration is a single segment of length $2 \times l$ and width w. In the first step, the objects B_l and B_s create new membranes, so the picture of this configuration consists on three segments. The new membrane created by B_s does not contains objects $+$ or $-$ and then the corresponding segment is not rotated with respect to the segment that represents the skin. On the other hand, the membrane created by B_l contains one object $+$, so its segment will be rotated an angle δ with respect to its parent membrane.

Notice also that the evolution of the objects L_1 and E has modified the number of objects F and W in the skin, so in this new picture, the segment corresponding to the skin has length $4 \times l$ and width $2 \times w$.

Figure 1 shows the graphical representation of the first four configurations where we fix a bottom-up orientation and an angle δ of 15 degrees.

2.2 Stochastic Versus Non-deterministic P Systems

The non-determinism is one of the main features of P systems and the possibility of reaching different configurations leads us to consider different graphical representations in the evolution of a P system.

One possible way to formalize the probability of obtaining one or other configuration is via stochastic P systems. Several alternatives to incorporate randomness into membrane systems can be found in the literature (see [1, 7, 8] and the references therein). One of them is to associate each rule of the P system with a probability. Thus, to pass from a configuration of the system to the next one we apply to every object present in the configuration a rule chosen at random, according to those probabilities, among all the rules whose left–hand side coincides with the object[2].

For example, let us consider Π_2 the following restricted P system with membrane creation:

- The alphabet of objects is $O = \{F, W, B_l, B_s, B_r, L, L_1, E\}$.
- The initial membrane structure together with the initial multiset of objects is $[F^2\, W\, B_l\, B_s\, L_1\, E]$.
- The rules are:

$$
\begin{array}{ll}
B_l \xrightarrow{1/2} [+\, F\, W\, B_l\, B_s\, L\, E] & \qquad L \;\; \to L\, F \\
B_l \xrightarrow{1/2} [-\, F\, W\, B_l\, B_s\, L\, E] & \qquad L_1 \to L_1\, F^2 \\
B_r \xrightarrow{1/2} [+\, F\, W\, B_l\, B_s\, L\, E] & \qquad E \;\; \to E\, W \\
B_r \xrightarrow{1/2} [-\, F\, W\, B_l\, B_s\, L\, E] & \qquad B_s \to [F\, W\, B_l\, B_r\, L_1\, E]
\end{array}
$$

[2] This idea was presented in [11].

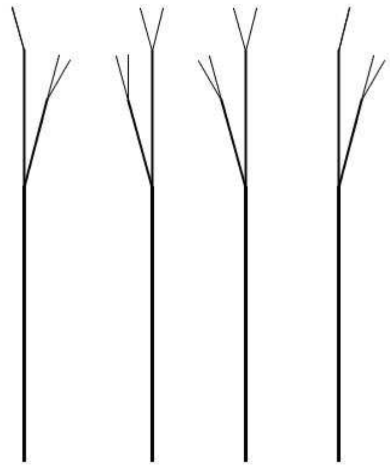

Fig. 2. Four configurations after the second step

There exist two rules for the evolution of the object B_l and two possibilities for the evolution of the object B_r. The probability for each choice is $1/2$. Notice that we do not make explicit the probability of the rule when this is one.

Figure 2 shows four different configurations after the second step of this P system with the angle $\delta = 15$.

3 JPLANT

In order to avoid the heavy hand-made computation for the graphical representation a new software tool has been designed. In this paper we present JPLANT [3], which computes the first configurations of a computation of a restricted P system with membrane creation and draws the corresponding graphical representation of the configurations of such computation.

JPLANT has been written in Java and it has a nice intuitive user-friendly graphical interface. The initial configuration and the set of rules are provided in plain text mode. The right syntax of the initial configuration and the rules are checked before starting the computation. The generation of a new configuration is driven by the user which can choose between jumping to a configuration N or generating (and drawing) at each time the next configuration.

The software tool is thought as a drawing tool so the computed new configurations are not showed to the user in the text mode. The output is a picture with a set of connected segments drawn according with the rules described in Section

[3] The software is available from [13].

2. For each new configuration, a new picture is drawn, so the output of this tool is a sequence of pictures which can be saved in several computer graphic formats.

The graphical representation of one configuration is not unique. It depends on the parameters l, w and δ which determine the length and width of the segments as well as the rotation angle with respect to the segment corresponding to the parent membrane. Such parameters are the input of the tool and they must be also provided by the user with the initial configuration and the rules.

The current version of JPLANT includes the ability of load and save files with the input data and save the generated pictures and also provide color to the pictures.

The color is one of the basic tools in the graphical design. In the current version, the color of the segment associated with each membrane is not associated with any object inside the membrane. In this way, we cannot change the color of a membrane by the analysis of the membrane structure of a computation. Nonetheless, JPLANT provides the ability of giving color to the generated picture. It is an ability which is not associated with the P system which generates the picture, but it is a powerful tool in order to get realistic representations.

4 Applications

Next we illustrate the possibilities of JPLANT with some examples.

4.1 Polygons and Spirals

Polygons and spirals can be considered a very special case of branching structures. They consists of a connected set of segments where a vertex only connect two segments. From a membrane computing point of view, this means that each membrane in a configuration only contains one membrane.

Polygons

A first example of figures built with P systems are regular polygons. In such polygons the length of the side is constant and the angle of deviation from the previous side is also constant. A simple calculus shows us that a deviation of $\delta = 360/n$ degrees allows us to built a regular polygon of n sides.

Figure 3 shows regular polygons of $n = 10$ and $n = 12$ sides obtained with $\delta = 36$ and $\delta = 30$ degrees. Obviously the number of steps are 10 and 12 respectively. The P system is the following

`Initial configuration:` $[F\,W\,H]$

`Rule:` $H \rightarrow [-F\,W\,H]$

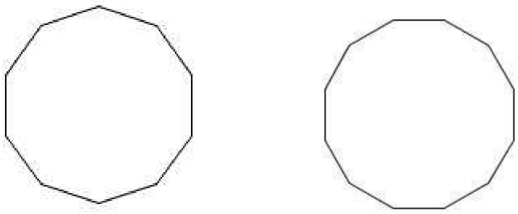

Fig. 3. 10-polygon and 12-polygon

Spirals

In mathematics, a spiral is a curve which emanates from a central point, getting progressively farther away as it revolves around the point. The concise mathematical definition is *the locus of a point moving at constant speed whose distance from a fixed point increases at a specific rate.*

An Archimedean spiral (a spiral named after the 3rd-century-BC Greek mathematician Archimedes) is the locus of points corresponding to the locations over time of a point moving away from a fixed point with a constant speed along a line which rotates with constant angular velocity. Equivalently, in polar coordinates (ρ, ω) it can be described by the equation $\rho = a + b\omega$ with real numbers a and b. Archimedes described such a spiral in his book *On Spirals*. It can be represented with the following P system:

`Initial configuration:` $[F^n WHL]$
`Rules:` $H \to [-F^n WLH]$ $L \to LF$

Figure 4 shows the representation of such Archimedes spiral for $n = 5$, length of $F = 0.01$, width $W = 1.0$, angle $\delta = 15$ and step 120.

The logarithmic spiral is a special kind of spiral curve which often appears in nature. It was first described by Descartes and extensively investigated by Jakob Bernoulli, who called it *Spira mirabilis*, "the marvelous spiral". Its equation in polar coordinates is $\rho = c^{\omega}$. It can be approximated by the P system

`Initial configuration:` $[F^n WHL]$
`Rules:` $H \to [-F^n WLH]$ $L \to LM_1F$ $M_1 \to M2$ $\ldots$ $M_{i-1} \to M_i$ $M_i \to L$

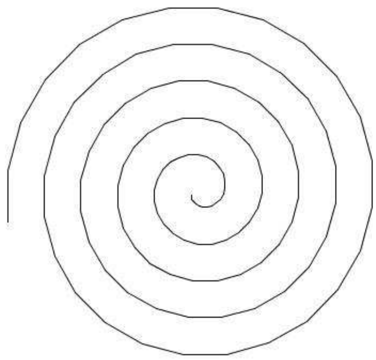

Fig. 4. Archimedes' spiral

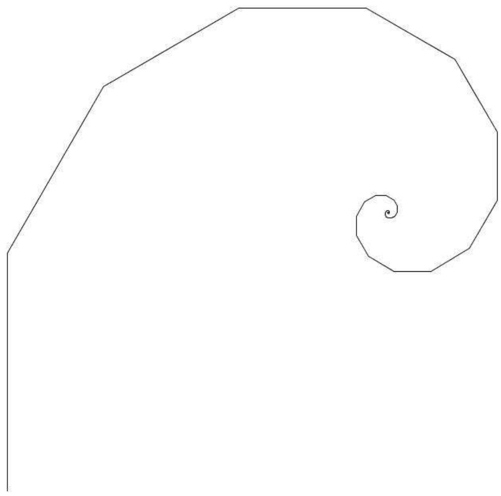

Fig. 5. Logarithmic spiral

Figure 5 shows the representation of such logarithmic spiral for $n = 10$, $i = 7$, length of $F = 0.001$, width $W = 1.0$ angle $\delta = 30$ and step 40.

4.2 Friezes

Another application of JPLANT for the graphical representation of restricted P systems with membrane creation is the design of friezes.

With the appropriate interpretation of the symbols, the following P system can be represented as a frieze based on right angles which has a flavor of Greek friezes. It can be extended horizontally in a non bounded way.

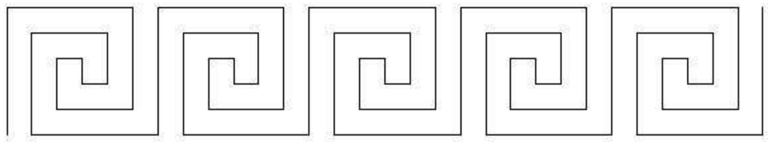

Fig. 6. The first frieze

`Initial configuration:` $[F^5\,W\,H_1]$	
`Rules:` $H_1 \rightarrow [-\,F^5\,W\,H_2]$	$H_7 \rightarrow [+\,F\,W\,H_8]$
$H_2 \rightarrow [-\,F^4\,W\,H_3]$	$H_8 \rightarrow [+\,F^2\,W\,H_9]$
$H_3 \rightarrow [-\,F^3\,W\,H_4]$	$H_9 \rightarrow [+\,F^3\,W\,H_{10}]$
$H_4 \rightarrow [-\,F^2\,W\,H_5]$	$H_{10} \rightarrow [+\,F^4\,W\,H_{11}]$
$H_5 \rightarrow [-\,F\,W\,H_6]$	$H_{11} \rightarrow [+\,F^5\,W\,H_{12}]$
$H_6 \rightarrow [-\,F\,W\,H_7]$	$H_{12} \rightarrow [+\,F^5\,W\,H_1]$

Figure 6 shows the representation of such frieze for length of $F = 0.5$, width $W = 1$ angle $\delta = 90$ and step 60.

Figure 7 shows a horizontally bounded frieze based on the Archimedes spiral.

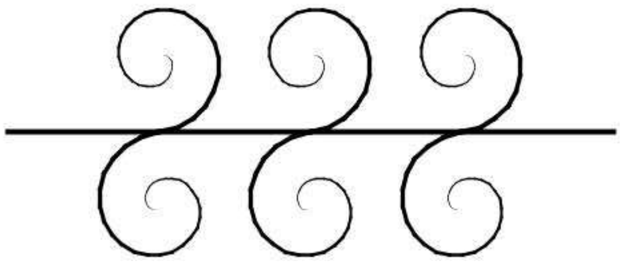

Fig. 7. The second frieze

`Initial configuration:` $[F^{300}\,W^{40}\,H_1\,I_1\,D_1]$	
`Rules:` $H_1 \rightarrow [F^{300}\,W^{40}\,H_2\,I_2\,D_2]$	$I_1 \rightarrow I_2$
$H_2 \rightarrow [F^{300}\,W^{40}\,H_3\,I_3\,D_3]$	$D_1 \rightarrow D_2$
$H_3 \rightarrow [F^{300}\,W^{40}]$	$I_2 \rightarrow I_3$
$L \rightarrow L\,F$	$D_2 \rightarrow D_3$
$K \rightarrow K\,W$	$I_3 \rightarrow I_4$
$I_4 \rightarrow [-^{11}\,F\,W\,L\,K\,D_4]$	$D_3 \rightarrow D_4$
$D_4 \rightarrow [+\,F\,W\,L\,K\,D_4]$	

Figure 7 shows the representation of such frieze for length of $F = 0.01$, width $W = 0.1$ angle $\delta = 15$ and step 40.

4.3 Plants

Figure 8 shows the corresponding graphical representation of the ninth configuration of the P system presented in Section 2.1, where we fix a bottom-up orientation with a length $F = 1$, width $W = 2$ and an angle δ of 15 degrees.

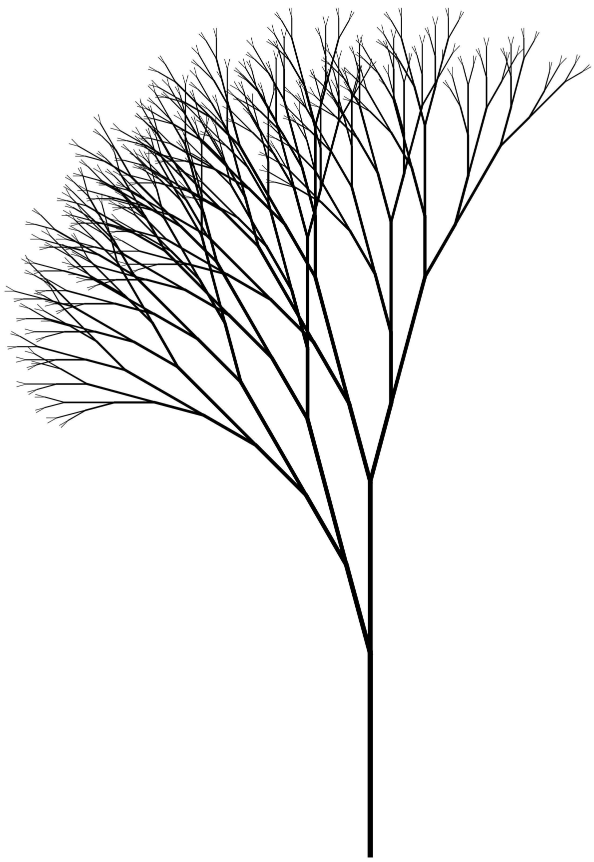

Fig. 8. Tree

Figure 9 represents four different trees obtained with JPLANT from the P system in Section 2.2.

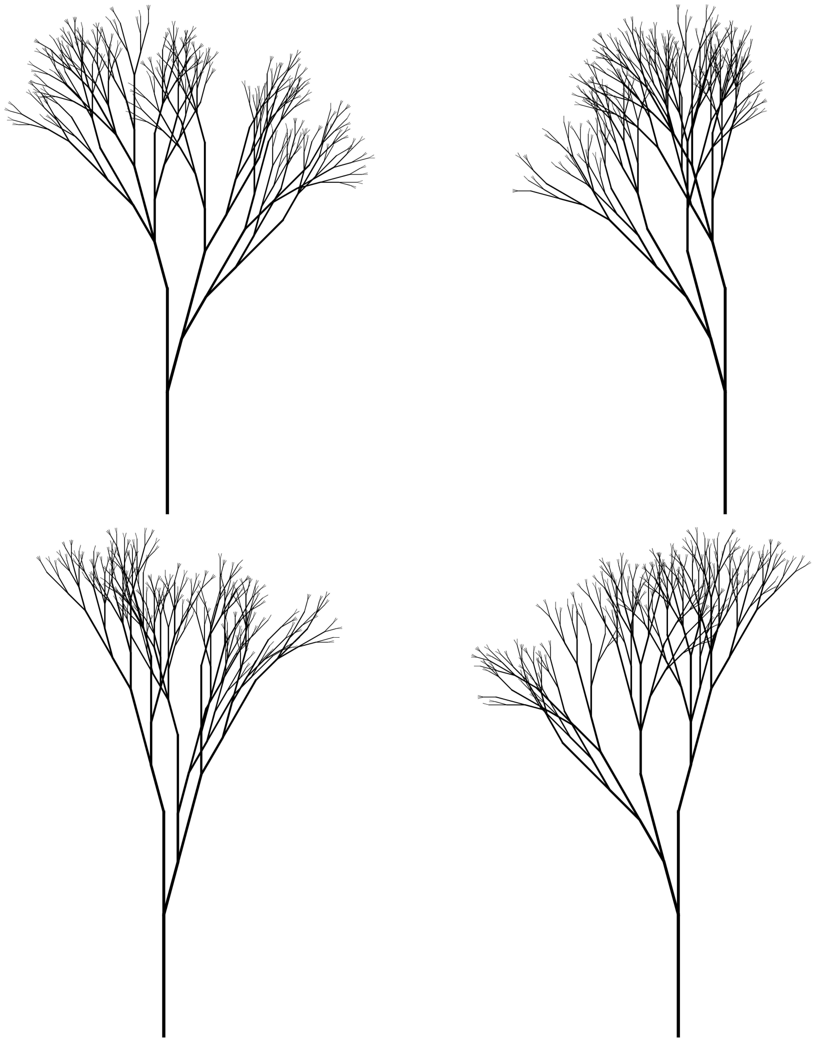

Fig. 9. Four configurations

5 Conclusions and Future Work

In this paper we have shown the suitability of P systems for modeling the growth of branching structures. It is our opinion that using membrane computing for this task could be an alternative to L-systems, the model most widely studied nowadays, for several reasons: the process of growing is closer to reality, since for example a plant does not grow by "rewriting" its branches, but by lengthening, widening and ramifying them; the membrane structure of P systems supports better and clearer the differentiation of the system into small units, easier to understand and possibly with different behaviors; the computational power of membrane systems can provide tools to easily simulate more complex models of growing, for example taking into account the flow of nutrients or hormones.

Nevertheless, it is still necessary a deeper study of several features of our proposed framework as compared with that of Lindenmayer systems. Two aspects that have to be investigated are the complexity of the models that can be constructed, and the computational efficiency in order to generate their graphical representation. On one hand, the use of the ingredients of membrane computing can lead to more intuitive models; on the other hand, we lose the linear sequence of graphical commands that characterize the parsing algorithm of L-systems.

From a theoretical point of view, one of the main drawbacks of the model is that it is extremely simple. Although the orientation of the paper belongs to the framework of membrane computing, the exclusive use of rules of type $a \to v$ and $a \to [v]$ miss the potential richness of expressiveness and computation of P systems. The following steps on this line should be devoted to the study of the graphical possibilities of P systems with more features, such as labels for the membranes (they can help to distinguish between different parts of a plant), the use of communication rules, allowing objects to cross the membranes of the system, division and/or dissolution rules, rules with cooperation, etc.

Acknowledgement

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

References

1. Ardelean, I.I., Cavaliere, M.: Modelling Biological Processes by Using a Probabilistic P System Software. Natural Computing, 2(2), (2003), 173–197.
2. Georgiou, A., Gheorghe, M.: Generative Devices Used in Graphics. In Alhazov, A., Martín–Vide, C., Păun, Gh. (eds.): Preproceedings of the Workshop on Membrane Computing. Technical Report, Vol. 28/03. Research Group on Mathematical Linguistics, Universitat Rovira i Virgili, Tarragona, (2003), 266–272.
3. Georgiou, A., Gheorghe, M., Bernardini, F.: Membrane–Based Devices Used in Computer Graphics. In Ciobanu, G. Păun, Gh., Pérez–Jiménez, M.J. (eds.): Applications of Membrane Computing. Springer–Verlag, Berlin Heidelberg New York, (2006), 253–282.
4. Ito, M., Martín–Vide, C., Păun, Gh.: A Characterization of Parikh Sets of ET0L Languages in Terms of P Systems. In Ito, M., Păun, Gh., Yu, S. (eds.): Words, Semigroups, and Transducers. World Scientific, (2001), 239–254.
5. Lindenmayer, A.: Mathematica Models for Cellular Interaction in Development, Parts I and II. Journal of Theoretical Biology, 18, (1968), 280–315.

6. Madhu, M., Krithivasan, K.: P Systems with Membrane Creation: Universality and Efficiency. In Margenstern, M., Rogozhin, Y. (eds.): Proceedings of the Third International Conference on Universal Machines and Computations. Lecture Notes in Computer Science, 2055, (2001), 276–287.
7. Obtulowicz, A., Păun, Gh.: (In Search of) Probabilistic P Systems. Biosystems 70(2), (2003), 107–121.
8. Pescini, D., Besozzi, D., Mauri, G., Zandron, C.: Dynamical Probabilistic P Systems. International Journal of Foundations of Computer Science 17(1), (2006), 183–204.
9. Păun, Gh.: Membrane Computing – An Introduction. Springer-Verlag, Berlin Heidelberg New York (2002).
10. Romero-Jiménez, A., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M. J.: The Growth of Branching Structures with P Systems. In Graciani–Díaz, C., Păun, Gh., Romero–Jiménez, A., Sancho–Caparrini, F. (eds.): Proceedings of the Fourth Brainstorming Week on Membrane Computing, Vol. II. Fénix Editora, Sevilla, (2006), 253–265.
11. Romero-Jiménez, A., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: Graphical Modelling of Higher Plants Using P Systems In H.J. Hoogeboom, Gh. Păun, G. Rozenberg, A. Salomaa (eds.) Membrane Computing, Seventh International Workshop. WMC 2006, Lecture Notes in Computer Science, 4361, (2006), 496-506.
12. Smith, A.R.: Plants, Fractals and Formal Languages. Computer Graphics, 18(3), (1984), 1–10.
13. The P Systems webpage `http://ppage.psystems.eu/`