A software tool for generating graphics by means of P systems

Elena Rivero-Gil · Miguel Á. Gutiérrez-Naranjo · Álvaro Romero-Jiménez · Agustín Riscos-Núñez

Abstract 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.

Keywords Membrane computing · P systems · Graphical representation · Software

1 Introduction

Since Lindenmayer systems (Lindenmayer 1968), best known as L-systems, were proposed (Smith 1984) as a tool for synthesizing realistic images of plants, many efforts have been done for bridging the theory of formal languages and computer graphics.

The first membrane-based device for computer graphics presented was a hybrid model between L-systems and membrane computing (Georgiou and Gheorghe 2003; Georgiou et al. 2006). It used concepts very close to the L-systems model. Later, a new approach was consider for representing the development of higher plants with P systems (Romero-Jiménez et al. 2006a). 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 of membranes in a P system with membrane creation.

E. Rivero-Gil · M. Á. Gutiérrez-Naranjo · Á. Romero-Jiménez (✉) · A. Riscos-Núñez
Departamento de Ciencias de la Computación e Inteligencia Artificial, Escuela Técnica Superior de Ingeniería Informática, Universidad de Sevilla, Avda. Reina Mercedes, s/n., 41012 Sevilla, Spain
e-mail: romero.alvaro@us.es

M. Á. Gutiérrez-Naranjo
e-mail: magutier@us.es

A. Riscos-Núñez
e-mail: ariscosn@us.es

By definition, the structure of membranes in a cell-like P system is a (formal) 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 hence 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 way similar to real plants. Romero-Jiménez et al. (2006b) completed this model by adding stochastic rules to the P system. In this case, the random choice of different rules produces different configurations of the P system and, hence, different graphical representations.

The hand-made graphical representation of the configurations of the P system becomes a hard task when the number of membranes and objects increases. For this reason, to study in depth the relationship between P systems and computer graphics it was necessary to develop a software able to deal with complex P systems and to represent graphically their evolution along time.

In this paper we present such a software, JPLANT, which computes the first configurations of a computation and draws the corresponding graphical representations. 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: Sect. 2 recalls 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. Details and updated information about P systems can be found in Păun (2002) and the P systems web page (http://ppage.psystems.eu/).

In this paper we consider P systems which make use of membrane creation rules (Ito et al. 2001; Madhu and Krithivasan 2001). 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 $\Pi = (O, \mu, w_1, \ldots, w_m, R)$ where:

(1) O is the alphabet of *objects*. There exist four distinguished objects, F, W, $+$ and $-$, that always belong to the alphabet. These objects have a graphical interpretation that will be explained 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) $w_1, \ldots, w_m$ 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* shared by each and every membrane, whose elements 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.
(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. If the system reaches a configuration where no rule can be applied, then the system halts.

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 branching structures, where a one-to-one identification between each node of the branching structure and each membrane of the P system is made.

A simple approach to graphically represent a membrane structure is to make a depth-first search of it. For each membrane a segment is drawn:

- The length of the segment is $m \times l$, where m is the multiplicity of F in the membrane and l is a fixed parameter whose value establishes the unit of length for the segments. Thus, if the number of copies of F in a membrane increases (resp. decreases) along the computation, the interpretation is that the corresponding segment is lengthening (resp. shortening)
- Analogously, the width of the segment is $n \times w$, where n is the multiplicity of W in the membrane and w is a fixed parameter whose value establishes the unit of width for the segments.
- Besides, each segment is drawn rotated with respect to the segment corresponding to its parent membrane. The rotation angle will be $k \times \delta$, where k is the multiplicity of objects "$+$" minus the multiplicity of objects "$-$" in the membrane, and δ is a fixed parameter whose value establishes the unit of angle for the rotation of the segments. This way, each object "$+$" means that the rotation angle is increased by δ whereas each object "$-$" means that it is decreased by δ.

The three magnitudes l, w and δ are exclusively associated with the representation of one configuration of the P system, and not with the system itself. They must be declared before the picture is depicted.

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

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

- 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:

$$\begin{array}{ll} B_l \rightarrow [+FWB_lB_sLE] & L \rightarrow LF \\ B_s \rightarrow [FWB_lB_rL_1E] & L_1 \rightarrow L_1F^2 \\ B_r \rightarrow [-FWB_lB_sLE] & E \rightarrow EW \end{array}$$

In this system an object B_s represents a straight branch 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 any graphical interpretation; they can be considered as seeds for making the branch grow in length and width.

The initial configuration consists of one membrane which contains two copies of F and one copy of W. Let us fix some values for 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 membrane. 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 membrane, so in this new picture, the segment corresponding to the skin membrane 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°.

2.2 Stochastic versus non-deterministic P systems

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 another configuration is via stochastic P systems. Several alternatives to incorporate randomness into membrane systems can be found in the literature (Ardelean and Cavaliere 2003; Obtulowicz and Păun 2003; Pescini et al. 2006). One of them is to associate each rule of the P system with a probability of execution. 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 (Romero-Jiménez et al. 2006b).

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 $[F^2\ W\ B_l\ B_s\ L_1\ E]$.

Fig. 1 Graphical representation of the first four configurations of the P system introduced as an example in Sect. 2.1. The orientation is fixed bottom-up and the value of the parameter δ is 15°

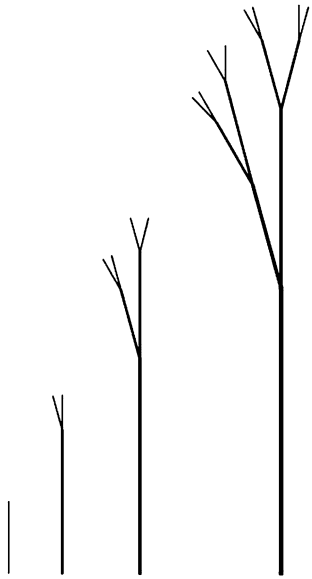
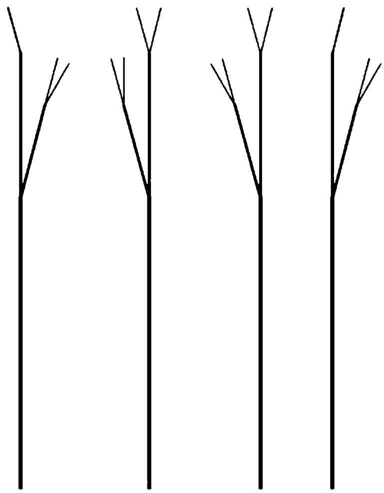


Fig. 2 Graphical representation of four possible configurations after the second step of the P system introduced as an example in Sect. 2.2. The orientation is fixed bottom-up and the value of the parameter δ is 15°



– The rules are:

$$\begin{array}{lc} B_l \rightarrow^{1/2} [+FWB_lB_sLE] & L \rightarrow LF \\ B_l \rightarrow^{1/2} [-FWB_lB_sLE] & L_1 \rightarrow L_1F^2 \\ B_r \rightarrow^{1/2} [+FWB_lB_sLE] & E \rightarrow EW \\ B_r \rightarrow^{1/2} [-FWB_lB_sLE] & B_s \rightarrow [FWB_lB_rL_1E] \end{array}$$

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

Figure 2 shows four different configurations that can be reached after the second step of this P system, where we fix a bottom-up orientation and an angle δ of 15°.

3 JPLANT

In order to avoid the heavy task of drawing by hand the graphical representation of a configuration of a restricted P system with membrane creation, a new software tool has been developed. In this section we present JPLANT, which computes the first configurations of a computation of such a P system and draws the corresponding graphical representations of these configurations. This software is available from the P systems web page (http://ppage.psystems.eu/).

JPLANT has been written in Java and it has an intuitive user-friendly graphical interface. The initial configuration and the set of rules are provided in plain text mode. The syntax of the initial configuration and the rules of the system are checked for correctness before starting the computation. The generation of a new configuration is driven by the user which can choose between jumping to a specific configuration 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 shown to the user in text mode. The output is a picture with a set of connected segments drawn according with the rules described in Sect. 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 angles between them. Such parameters must also be provided by the user as input to the tool, together with the initial configuration and the rules.

The current version of JPLANT includes the ability to load and save files with the input data and to save the generated pictures. Another feature of JPLANT is the possibility of coloring the graphics generated from the configurations of the system. This is performed by drawing the segments associated with each membrane in different colors, obtaining this

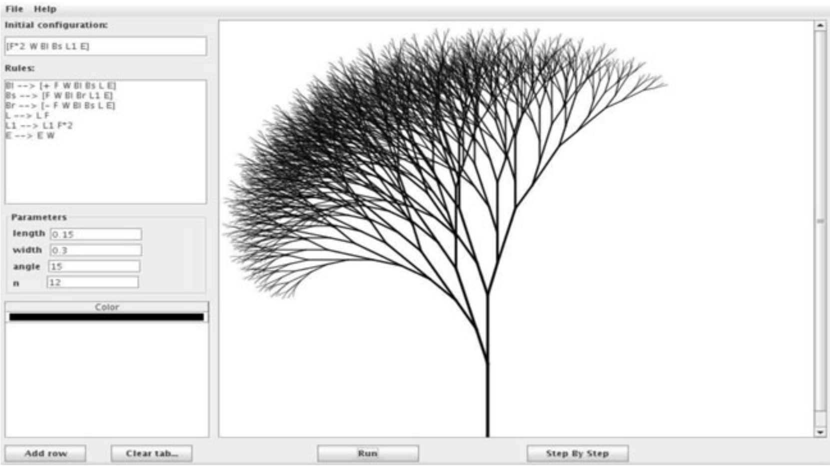


Fig. 3 Snapshot of the GUI of the software JPLANT. It can be observed that the initial configuration, the rules, the values of the parameters and the number of computation steps can be introduced to obtain the graphical representation of one configuration

way more realistic pictures. We must point out that these graphical elements are not intrinsic to the P system, but are indicated externally to it.

Figure 3 shows a snapshot of the graphical user interface of this computer software.

4 Applications

Next we illustrate the possibilities of JPLANT with some geometrical examples. We begin with polygons and spirals, which can be considered a very special case of branching structures. Indeed they consist of a connected set of segments where each vertex only connects two segments. Next, we deal with friezes, that is, structures which are repetitive in one direction. We conclude the section by discussing some examples related to higher plants.

4.1 Polygons and spirals

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 $\delta = 360/n°$ allows us to build a regular polygon of n sides.

Figure 4 shows regular polygons of $n = 10$ and $n = 12$ sides obtained with $\delta = 36$ and $\delta = 30°$, respectively. Obviously the number of steps are 10 for the first polygon and 12 for the second one. The P system is the following:

Initial configuration:	$[FWH]$
Rule:	$H \rightarrow [-FWH]$

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:

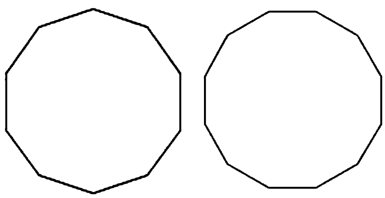


Fig. 4 Graphical representation of the 10th and 12th configuration of a P system able to represent regular polygons. The values of the parameter δ are 36 and 30°, respectively

```
Initial configuration:   [F^i WLH]
               Rules:   H → [−F^i WLH]
                        L → LF
```

Figure 5 shows the representation of the 120th configuration of the system representing the Archimedes spiral, with values of the parameters $l = 0.01$, $w = 1$ and $\delta = 15°$ (and the value of the technical parameter $i = 5$).

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^i WLH]
               Rules:   H → [−F^i WLH]
                        L → LM_1 F
                        M_1 → M_2
                        ...
                        M_{j−1} → M_j
                        M_j → L
```

Figure 6 shows the representation of the 40th configuration of the system representing the logarithmic spiral, with values of the parameters $l = 0.001$, $w = 1$ and $\delta = 30°$ (and the value of the technical parameters $i = 10$ and $j = 7$).

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 configurations of the following P system can be graphically represented as a frieze based on right angles which has a flavor of Greek friezes. It can be extended horizontally in an unbounded manner.

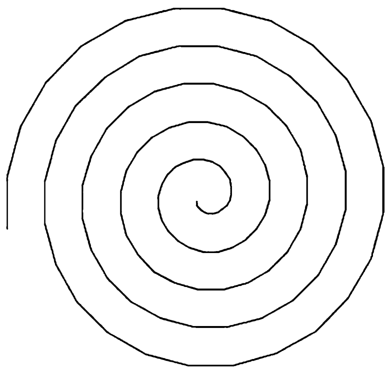


Fig. 5 Graphical representation of the 120th configuration of the system representing the Archimedean spiral, with values of the parameters $l = 0.01$, $w = 1$ and $\delta = 15°$

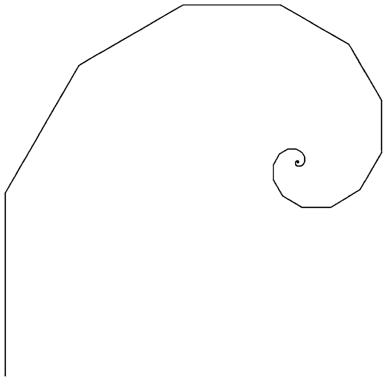


Fig. 6 Graphical representation of the 40th configuration of the system representing the logarithmic spiral, with values of the parameters $l = 0.001$, $w = 1$ and $\delta = 30°$

`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 7 shows the representation of the 60th configuration of the system representing such frieze, with values of the parameters $l = 0.5$, $w = 1$ and $\delta = 90°$

Combining ideas from the previous examples, a P system representing a horizontally bounded frieze based on the Archimedes spiral can be constructed.

`Initial configuration:` $[F^{300}\, W^{40}\, H_1\, I_1\, D_1]$

`Rules:`

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

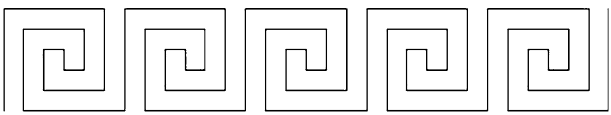


Fig. 7 Graphical representation of the 60th configuration of the system representing a first example of frieze, with values of the parameters $l = 0.5$, $w = 1$ and $\delta = 90°$

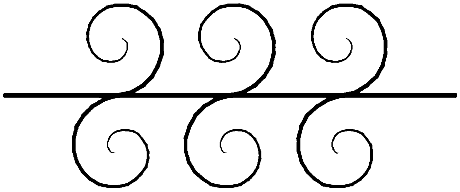


Fig. 8 Graphical representation of the 40th configuration of the system representing a second example of frieze, with values of the parameters $l = 0.01$, $w = 0.1$ and $\delta = 15°$

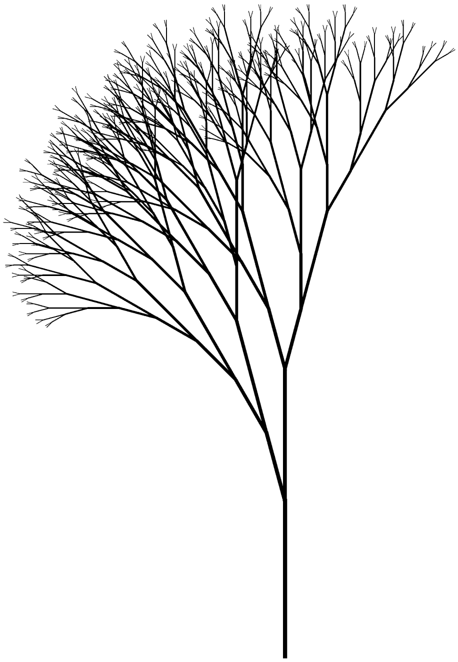


Fig. 9 Graphical representation of the 9th configuration of the restricted P system with membrane creation presented in Sect. 2.1

Figure 8 shows the representation of the 40th configuration of such system, with values of the parameters $l = 0.01$, $w = 0.1$ and $\delta = 15°$.

4.3 Higher plants

Figure 9 shows the graphical representation of the ninth configuration of the P system presented in Sect. 2.1, which is a first example presented of a restricted P system with membrane creation. The orientation is bottom-up and the values of the parameters are $l = 1$, $w = 2$ and $\delta = 15°$.

Figure 10 shows the graphical representation of the ninth step of four different computations that the P system presented in Sect. 2.2 can perform. This is a first example of a stochastic restricted P system with membrane creation. The orientation is bottom-up and the values of the parameters are $l = 1$, $w = 2$ and $\delta = 15°$.

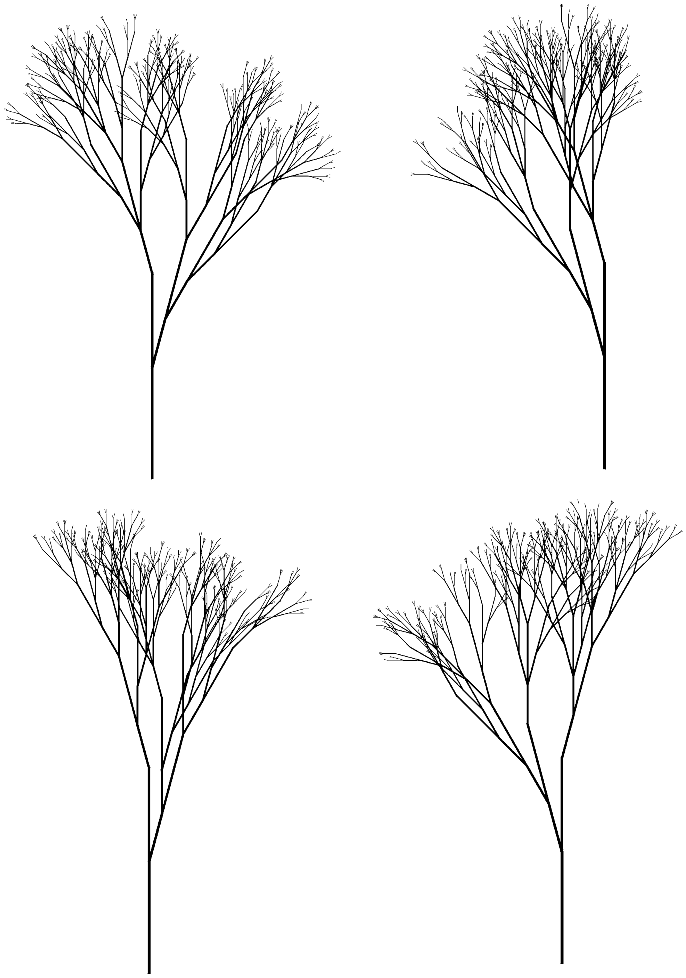


Fig. 10 Four different trees that can be obtained after carrying out 9 computation steps of the stochastic restricted P system with membrane creation presented in Sect. 2.2

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 of restricted P systems with membrane creation 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 for example 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 and rules with cooperation.

Acknowledgments 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

Ardelean II, Cavaliere M (2003) Modelling biological processes by using a probabilistic P system software. Nat Comput 2(2):173–197

Georgiou A, Gheorghe M (2003) P systems used in plant graphics. In: Alhazov A, Martín-Vide C, Păun Gh (eds) Preproceedings of the workshop on membrane computing. Tarragona, Spain

Georgiou A, Gheorghe M, Bernardini F (2006) Membrane-based devices used in computer graphics. In: Ciobanu G, Păun Gh, Pérez Jiménez MJ (eds) Applications of membrane computing. Springer, Berlin

Ito M, Martín-Vide C, Păun Gh (2001) A characterization of parikh sets of ET0L languages in terms of P systems. In: Ito M, Păun Gh, Yu S (eds) Words, semigroups, and transductions. World Scientific, Singapore

Lindenmayer A (1968) Mathematical models for cellular interaction in development, parts I and II. J Theor Biol 18:280–315

Madhu M, Krithivasan K (2001) P systems with membrane creation: universality and efficiency. In: Margenstern M, Rogozhin Y (eds) MCU 2001: machines, computations, and universality. Third international conference, Chişinău, Moldova, May 2001. Lecture notes in computer science, vol 2055. Springer, Berlin, pp 276–287

Obtulowicz A, Păun Gh (2003) (In search of) probabilistic P systems. BioSystems 70(2):107–121

Pescini D, Besozzi D, Mauri G, Zandron C (2006) Dynamical probabilistic P systems. Int J Found Comput Sci 17(1):183–204

Păun Gh (2002) Membrane computing. An introduction. Springer, Berlin

Romero-Jiménez Á, Gutiérrez-Naranjo MÁ, Pérez-Jiménez MJ (2006a) The growth of branching structures with P systems. In: Graciani Díaz C, Păun Gh, Romero Jiménez Á, Sancho Caparrini F (eds) Proceedings of the fourth brainstorming week on membrane computing, vol II. Sevilla, Spain, 2006

Romero-Jiménez Á, Gutiérrez-Naranjo MÁ, Pérez-Jiménez MJ (2006b) Graphical modelling of higher plants using P systems. In: Hoogeboom HJ, Păun Gh, Rozenberg G, Salomaa A (eds) WMC 2006: membrane computing. Seventh international workshop, Leiden, The Netherlands, July 2006. Lecture notes in computer science, vol 4361. Springer, Berlin, pp 496–506

Smith AR (1984) Plants, fractals and formal languages. Comput Graph 18(3):1–10

The P Systems Web Page (2009) http://ppage.psystems.eu/