P Systems based Computing Polynomials: Design and Formal Verification

Weitao Yuan[1], Gexiang Zhang[1⋆], Mario J. Pérez-Jiménez[2], Tao Wang[1], and Zhiwei Huang[1]

[1]School of Electrical Engineering, Southwest Jiaotong University, Chengdu, 610031, P.R. China
email: 657279959@qq.com, zhgxdylan@126.com
[2]Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
Universidad de Sevilla, Sevilla, 41012, Spain
email: marper@us.es

Abstract. Automatic design of P systems is an attractive research topic in the community of membrane computing. Differing from the previous work that used evolutionary algorithms to fulfill the task, this paper presents the design of a simple (deterministic transition) P system (without input membrane) of degree 1, capturing the value of the k-order ($k \geq 2$) polynomial by using a reasoning method. Specifically, the values of polynomial $p(n)$ corresponding to a natural number t is equal to the multiplicity of a distinguished object of the system (the output object) in the configuration at instant t. We also discuss the descriptive computational resources required by the designed k-order polynomial P system.
Keywords: Membrane computing, P system, automatic design, polynomial.

1 Introduction

Since the area of membrane computing was initiated in 1998 [1], the rapidly theoretical development with respect to computing models and their computing power and computational efficiency [2] and various real-world applications [3, 4] have been reported. A P system for performing a specific task, especially for solving an NP-hard, NP-complete or PSPACE-complete problem or for controlling robots, is carefully designed by experts and cannot be automatically gained by using programs, which may limit the application of P systems. How to automatically design a P system by using programs, namely, the programmability of a P system, is an attractive research direction in the area of membrane computing [5].

The automatic design of a P system is a very complicated and challenging task. The related work in the literature focused on the use of evolutionary algorithms to make a population of P systems evolve toward a successful one. This

⋆ Corresponding author.

work started with the selection of an appropriate subset from a redundant set of evolution rules to design a cell-like P system, where a membrane structure and initial objects were pre-defined and fixed in the process of design [6–9, 4]. In [6], a genetic algorithm was used to design a P system to calculate 4^2. In [7], a binary encoding technique was presented to denote an evolution rule set of a P system and a quantum-inspired evolutionary algorithm (QIEA) was used to make a population of P systems evolve toward successful ones. This method successfully solved the design of P systems to compute 4^2 and n^2 (for natural numbers $n \geq 2$). In [8], an evaluation approach considering non-determinism and halting penalty factors and a genetic algorithm with the binary encoding technique in [7] were introduced to design P systems for 4^2, n^2 and the generation of language $\{a^{2^n}b^{3^n}|n > 1\}$. In these studies mentioned above, a specific redundant evolution rule set was designed for a specific computational task. This was developed in [4, 9] by applying one pre-defined redundant evolution rule set to design multiple different P systems, each of which executes a computation task. In [9], an automatic design method of a cell-like P system framework for performing five basic arithmetic operations (addition, subtraction, multiplication, division and power) was presented. In [4], a common redundant set of evolution rules was applied to design successful P systems for fulfilling eight computational tasks: $2(n-1)$, $2n-1$, n^2, $\frac{1}{2}[(n-1)^2+(n-1)]$, $(n-1)^2+(n-1)$, $(n-1)^2+2^n+2$, $a^{2^n}b^{3^n}$ and $\frac{1}{2}(3^n-1)$, ($n > 1$ or 2). A significant development in this topic is the work in [10] in which a cell-like halting P system for 4^2 was designed by tuning membrane structures, initial objects and evolution rules. In this work, a genetic algorithm with a binary encoding technique was discussed to codify the membrane structure, initial objects and evolution rules of a P system. Following this work, an automatic design method, *Permutation Penalty Genetic Algorithm* (PPGA), for a deterministic and non-halting membrane system by tuning membrane structures, initial objects and evolution rules was proposed in [5]. The main ideas of PPGA are the introduction of the permutation encoding technique for a membrane system, a penalty function evaluation approach for a candidate membrane system and a genetic algorithm for making a population of P systems evolve toward a successful one fulfilling a given computational task. A cell-like membrane system for computing the square of n^2 (for natural numbers $n \geq 1$) was successfully designed. In addition, the automatic design of the minimal membrane systems with respect to their membrane structures, alphabet, initial objects and evolution rules to fulfill the given task were also discussed in [5].

On the basis of the studies mentioned above, our aim is to design a P system to compute a general polynomial. This paper proposes a reasoning method to implement the design of a simple (deterministic transition) P system (without input membrane) of degree 1, capturing the value of the k-order polynomial, and it also presents the descriptive computational resources required by the designed k-order polynomial P system. The reasoning method is the use of the characteristics of P systems with a membrane structure, initial objects and rules, and the semantics of the P systems to provide a way to design the required

membrane systems. The introduced technique is completely different from the previous work that used evolutionary algorithms to design a P system.

The remainder of this paper is organized as follows: Section 2 describes the polynomials we aim to design. In Sections 3, the two-, three- and k-order polynomial P systems are designed. Section 4 discusses the descriptive computational resources that the two-, three- and k-order polynomial P systems require. Section 5 concludes this work.

2 Polynomials computed by P systems

We start by defining what computing a k-order polynomial means in the framework of deterministic transition P system without input membrane.

Definition 1. *Let $p(n)$ be a polynomial whose coeficients are natural numbers. We say that $p(n)$ is computed by a deterministic transition P system without input membrane $\Pi_{p(n)} = (\Gamma, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}_1, \ldots, \mathcal{R}_q, i_{out})$, if the following holds:*

- *There exists a distinguished object (the output object), $\mathbf{o}$, in the working alphabet Γ.*
- *For each $t \in \mathbf{N}$, in the configuration $\mathcal{C}_t$ of $\Pi_{p(n)}$ at instant t, the multiplicity of object $\mathbf{o}$ in the output membrane labelled by i_{out} is $p(t)$.*

From the previous definition, if a deterministic transition P system without input membrane computes a polynomial, then the computation of that system is a non-halting computation.

We try to find a minimum such P system computing a k-order polynomial. Here the concept "minimum" refers to the following constraints associated with P systems:

- The membrane structure has only one membrane.
- The evolution rules are non-cooperative.
- The number of objects and rules must be minimum.

3 Polynomial P systems design

3.1 Two-order polynomial P system design

In this section we present a (deterministic) transition P system (without input membrane), $\Pi_{p(n)}$, of degree 1 that computes the arbitrary polynomial with order 2, $p(n) = a_2 \cdot n^2 + a_1 \cdot n + a_0$, where a_0, a_1, a_2 are natural numbers, in the sense of Definition 1.

It is worth pointing out that $p(n+1) = p(n) + 2a_2 \cdot n + a_2 + a_1$.

For each symbol x, the expression x^0 denotes the empty multiset (or the empty string).

Associated with the polynomial $p(n) = a_2 \cdot n^2 + a_1 \cdot n + a_0$, we consider the (deterministic) transition P system (without input membrane) $\Pi_{p(n)} = (\Gamma, \mu, \mathcal{M}_1, \mathcal{R}_1, i_{out})$ of degree 1, defined as follows:

- $\Gamma = \{b_1, b_2, \mathbf{o}\}$ and the output object is $\mathbf{o}$;
- $\mu = [\ ]_1$;
- $\mathcal{M}_1 = \{\mathbf{o}^{a_0}\, b_1\}$;
- R_1 is the set of the following non-cooperative evolution rules:
  - $r_1 \equiv [\, b_1 \longrightarrow \mathbf{o}^{a_2+a_1}\, b_1\, b_2\,]_1$;
  - $r_2 \equiv [\, b_2 \longrightarrow \mathbf{o}^{2a_2}\, b_2\,]_1$.
- $i_{out} = 1$.

Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_t, \ldots)$ be the (unique) computation of Π. For each $t \in \mathbf{N}$ and $x \in \Gamma$, we denote by $\mathcal{C}_t(1)$ the multiset over Γ associated with the membrane 1 in configuration $\mathcal{C}_t$.

Theorem 1. *For every $t \in \mathbf{N}$ we have $\mathcal{C}_t(1) = \{\mathbf{o}^{p(t)}\, ,\, b_1\, ,\, {b_2}^t\}$.*

Proof. Let us prove the result by induction on t.

For the base case, $t = 0$, we note that $\mathcal{C}_0(1) = \mathcal{M}_1 = \{\mathbf{o}^{a_0}\, b_1\} = \{\mathbf{o}^{p(0)}\, ,\, b_1\, ,\, {b_2}^0\}$.

Let us assume the result holds for $t \geq 0$, that is, let us suppose that $\mathcal{C}_t(1) = \{\mathbf{o}^{p(t)}\, ,\, b_1\, ,\, {b_2}^t\}$. In order to obtain $\mathcal{C}_{t+1}(1)$, we must apply the rules r_1 and r_2 to configuration $\mathcal{C}_t$. Then

$$\begin{aligned}\mathcal{C}_{t+1}(1) &= \{\mathbf{o}^{p(t)}\, ,\, \mathbf{o}^{a_2+a_1}\, , b_1\, , b_2\, ,\, \mathbf{o}^{2a_2 \cdot t}\, , {b_2}^t\, \} \\ &= \{\mathbf{o}^{p(t)+2a_2 \cdot t+a_2+a_1}\, ,\, b_1\, , {b_2}^{(t+1)}\, \} \\ &= \{\mathbf{o}^{p(t+1)}\, ,\, b_1\, , {b_2}^{(t+1)}\, \}\end{aligned}$$

Hence, the result holds for $t + 1$.

Corollary 1. *For every $t \in \mathbf{N}$, in configuration $\mathcal{C}_t$ of $\Pi_{p(n)}$ at instant t, the multiplicity of object $\mathbf{o}$ in the membrane of the system is $p(t)$.*

3.2 Three-order polynomial P system design

In this section we present a (deterministic) transition P system (without input membrane), $\Pi_{p(n)}$, of degree 1 that computes the arbitrary polynomial with order 3,

$$p(n) = a_3 \cdot n^3 + a_2 \cdot n^2 + a_1 \cdot n + a_0,$$

where a_0, a_1, a_2, a_3 are natural numbers, in the sense of Definition 1.

It is worth pointing out that $p(n+1) = p(n)+3a_3n^2+(3a_3+2a_2)n+a_3+a_2+a_1$.

Associated with the polynomial $p(n) = a_3 \cdot n^3+a_2 \cdot n^2+a_1 \cdot n+a_0$, we consider the (deterministic) transition P system (without input membrane), $\Pi_{p(n)} = (\Gamma, \mu, \mathcal{M}_1, R_1)$, of degree 1 defined as follows:

- $\Gamma = \{b_1, b_2, b_3, \mathbf{o}\}$ and the output object is $\mathbf{o}$;
- $\mu = [\ ]_1$;
- $\mathcal{M}_1 = \{\mathbf{o}^{a_0}\, b_1\}$;
- R_1 is the set of the following non-cooperative evolution rules:
  - $r_1 \equiv [\, b_1 \longrightarrow \mathbf{o}^{a_3+a_2+a_1}\, b_1\ b_2\, b_3\,]_1$;
  - $r_2 \equiv [\, b_2 \longrightarrow \mathbf{o}^{3a_3+2a_2}\, b_2\, {b_3}^2\,]_1$;
  - $r_3 \equiv [\, b_3 \longrightarrow \mathbf{o}^{3a_3}\, b_3\,]_1$.

Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_t, \ldots)$ be the (unique) computation of Π. For each $t \in \mathbf{N}$ and $x \in, \Gamma$ we denote by $\mathcal{C}_t(1)$ the multiset over Γ associated with membrane 1 in configuration $\mathcal{C}_t$.

Theorem 2. *For every $t \in \mathbf{N}$ we have $\mathcal{C}_t(1) = \{\mathbf{o}^{p(t)}\, ,\, b_1\, ,\, {b_2}^t\, ,\, {b_3}^{t^2}\}$.*

Proof. Let us prove the result by induction on t. For the base case, $t = 0$, we note that $\mathcal{C}_0(1) = \mathcal{M}_1 = \{\mathbf{o}^{a_0}\, b_1\} = \{\mathbf{o}^{p(0)}\, ,\, b_1\, ,\, {b_2}^0\, ,\, {b_3}^{0^2}\}$.

Let us assume the result holds for $t \geq 0$, that is, let us suppose that $\mathcal{C}_t(1) = \{\mathbf{o}^{p(t)}\, ,\, b_1\, ,\, {b_2}^t\, ,\, {b_3}^{t^2}\}$. In order to obtain $\mathcal{C}_{t+1}(1)$, we can apply the rules r_1, r_2 and r_3 to configuration $\mathcal{C}_t$. Then

$$\begin{aligned}\mathcal{C}_{t+1}(1) &= \{\mathbf{o}^{p(t)}\, ,\, \mathbf{o}^{a_3+a_2+a_1}\, ,\, b_1\ ,\, b_2\, ,\, b_3\, ,\, \mathbf{o}^{(3a_3+2a_2)\cdot t}\, ,\, {b_2}^t\, ,\, {b_3}^{2t}\, ,\, \mathbf{o}^{3a_3t^2}\, ,\, {b_3}^{t^2}\} \\ &= \{\mathbf{o}^{p(t)+3a_3t^2+(3a_3+2a_2)t+a_3+a_2+a_1}\, ,\, b_1\, ,\, {b_2}^{t+1}\, ,\, {b_3}^{t^2+2t+1}\} \\ &= \{\mathbf{o}^{p(t+1)}\, ,\, b_1\, ,\, {b_2}^{t+1}\, ,\, {b_3}^{(t+1)^2}\}\end{aligned}$$

Hence, the result holds for $t + 1$.

Corollary 2. *For every $t \in \mathbf{N}$, in configuration $\mathcal{C}_t$ of $\Pi_{p(n)}$ at instant t, the multiplicity of object $\mathbf{o}$ in the membrane of the system is $p(t)$.*

3.3 *k*-order polynomial P system design

In this section we present a (deterministic) transition P system (without input membrane), $\Pi_{p_k(n)}$, of degree $k \geq 1$ that computes the arbitrary polynomial with order k, $p_k(n) = a_k \cdot n^k + a_{k-1} \cdot n^{k-1} + \cdots + a_1 \cdot n^1 + a_0 \cdot n^0$, where $a_k, a_{k-1}, \ldots, a_1, a_0$ are natural numbers.

It is worth pointing out that the increment $p_k(n+1)-p_k(n)$ of polynomial $p_k(n)$ is:

$$\left[\binom{k}{1} a_k\right] \cdot n^{k-1} + \left[\binom{k}{2} a_k + \binom{k-1}{1} a_{k-1}\right] \cdot n^{k-2} +$$
$$\left[\binom{k}{3} a_k + \binom{k-1}{2} a_{k-1} + \binom{k-2}{1} a_{k-2}\right] \cdot n^{k-3} +$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$\left[\binom{k}{k-1} a_k + \binom{k-1}{k-2} a_{k-1} + \cdots + \binom{3}{2} a_3 + \binom{2}{1} a_2\right] \cdot n^1 +$$
$$\left[\binom{k}{k} a_k + \binom{k-1}{k-1} a_{k-1} + \cdots + \binom{2}{2} a_2 + \binom{1}{1} a_1\right] \cdot n^0$$

Let us denote:

- $\alpha_k^{(0)} = \binom{k}{k} a_k + \binom{k-1}{k-1} a_{k-1} + \cdots + \binom{2}{2} a_2 + \binom{1}{1} a_1$
- $\alpha_k^{(1)} = \binom{k}{k-1} a_k + \binom{k-1}{k-2} a_{k-1} + \cdots + \binom{3}{2} a_3 + \binom{2}{1} a_2$
- $\alpha_k^{(2)} = \binom{k}{k-2} a_k + \binom{k-1}{k-3} a_{k-1} + \cdots + \binom{4}{2} a_4 + \binom{3}{1} a_3$
- $\alpha_k^{(3)} = \binom{k}{k-3} a_k + \binom{k-1}{k-4} a_{k-1} + \cdots + \binom{5}{2} a_5 + \binom{4}{1} a_4$

  $\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$
- $\alpha_k^{(k-1)} = \binom{k}{1} a_k$

Then:

$$p_k(n+1) - p_k(n) = \alpha_k^{(0)} \cdot t^0 + \alpha_k^{(1)} \cdot t + \alpha_k^{(2)} \cdot t^2 + \cdots + \alpha_k^{(k-1)} \cdot t^{k-1}$$

Associated with polynomial $p_k(n) = a_k \cdot n^k + a_{k-1} \cdot n^{k-1} + \cdots + a_1 \cdot n^1 + a_0 \cdot n^0$ of order k, we consider the (deterministic) transition P system (without input membrane) $\Pi_{p_k(n)} = (\Gamma, \mu, \mathcal{M}_1, \mathcal{R}_1)$ of degree 1, defined as follows:

- $\Gamma = \{\mathbf{o}, b_1, b_2, \ldots, b_k\}$ and the output object is $\mathbf{o}$;
- $\mu = [\ ]_1$;
- $\mathcal{M}_1 = \{\mathbf{o}^{a_0}\, b_1\}$;
- R_1 is the set of the following non-cooperative evolution rules:
  - $r_1 \equiv [\, b_1 \longrightarrow \mathbf{o}^{\alpha_k^{(0)}}\, b_1^{\binom{0}{0}}\, b_2^{\binom{1}{0}}\, b_3^{\binom{2}{0}}\, b_4^{\binom{3}{0}}\, \ldots\, b_k^{\binom{k-1}{0}}]_1$;
  - $r_2 \equiv [\, b_2 \longrightarrow \mathbf{o}^{\alpha_k^{(1)}}\, b_2^{\binom{1}{1}}\, b_3^{\binom{2}{1}}\, b_4^{\binom{3}{1}}\, \ldots\, b_k^{\binom{k-1}{1}}]_1$;
  - $r_3 \equiv [\, b_3 \longrightarrow \mathbf{o}^{\alpha_k^{(2)}}\, b_3^{\binom{2}{2}}\, b_4^{\binom{3}{2}}\, \ldots\, b_k^{\binom{k-1}{2}}]_1$;
  - $r_4 \equiv [\, b_4 \longrightarrow \mathbf{o}^{\alpha_k^{(3)}}\, b_4^{\binom{3}{3}}\, \ldots\, b_k^{\binom{k-1}{3}}]_1$;

    $\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$

– $r_k \equiv [\, b_k \longrightarrow \mathbf{o}^{\alpha_k^{(k-1)}}\, b_k^{\binom{k-1}{k-1}}\,]_1$;

Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_t, \ldots)$ the (unique) computation of Π. For each $t \in \mathbf{N}$ and $x \in, \Gamma$ we denote by $\mathcal{C}_t(1)$ the multiset over Γ associated with the membrane 1 in configuration $\mathcal{C}_t$.

Theorem 3. *For every $t \in \mathbf{N}$, we have $\mathcal{C}_t(1) = \{\mathbf{o}^{p_k(t)}\, ,\, b_1\, ,\, {b_2}^t\, ,\, \ldots\, ,\, {b_k}^{t^{k-1}}\}$.*

Proof. Let us prove the result by induction on t.

For the base case, $t = 0$, let us notice that $\mathcal{C}_0(1) = \mathcal{M}_1 = \{\mathbf{o}^{a_0}\, b_1\} = \{\mathbf{o}^{p_k(0)}\, ,\, b_1, b_2^0, \ldots, {b_k}^0\}$.

Let us assume the result holds for $t \geq 0$, that is, let us suppose that

$$\mathcal{C}_t(1) = \{\mathbf{o}^{p_k(t)}\, ,\, b_1\, ,\, {b_2}^t\, ,\, \ldots\, ,\, {b_k}^{t^{k-1}}\}$$

In order to obtain $\mathcal{C}_{t+1}(1)$, we must apply the rules r_1, r_2 and r_k to configuration $\mathcal{C}_t$. Then

$$\begin{aligned}
\mathcal{C}_{t+1}(1) &= \{\mathbf{o}^{p_k(t)}, \mathbf{o}^{\alpha_k^{(0)}+\alpha_k^{(1)}\cdot t+\alpha_k^{(2)}\cdot t^2+\cdots+\alpha_k^{(k-1)}\cdot t^{k-1}}, b_1, {b_2}^{\binom{1}{0}+\binom{1}{1}\cdot t}, \\
&\quad {b_3}^{\binom{2}{0}+\binom{2}{1}\cdot t\binom{2}{2}\cdot t^2}, \ldots, {b_k}^{\binom{k-1}{0}+\binom{k-1}{1}\cdot t+\cdots+\binom{k-1}{k-1}\cdot t^{k-1}}\} \\
&= \{\mathbf{o}^{p_k(t)+\alpha_k^{(0)}+\alpha_k^{(1)}\cdot t+\alpha_k^{(2)}\cdot t^2+\cdots+\alpha_k^{(k-1)}\cdot t^{k-1}}, b_1, {b_2}^{(t+1)}, {b_3}^{(t+1)^2}, \ldots, \\
&\quad {b_k}^{(t+1)^{k-1}}\} \\
&= \{\mathbf{o}^{p_k(t+1)}, b_1, {b_2}^{(t+1)}, {b_3}^{(t+1)^2}, \ldots, {b_k}^{(t+1)^{k-1}}\}
\end{aligned}$$

Hence, the result holds for $t + 1$.

Corollary 3. *For every $t \in \mathbf{N}$, in configuration $\mathcal{C}_t$ of $\Pi_{p_k(n)}$ at instant t, the multiplicity of object $\mathbf{o}$ in the membrane of the system is $p_k(t)$.*

4 Descriptive computational resources

This section discusses the descriptive computational resources required by the designed two-, three- and k–order polynomial P systems. According to the design procedures of two- and three-order polynomial P systems, the amount of resources to build the designed P systems (of degree 1) in both cases are 2 rules and 3 objects, 3 rules and 4 objects, respectively. In what follows, we discuss the descriptive computational resources required by the designed k–order polynomial P system $\Pi_{p_k(n)}$ of degree 1.

– The size of the working alphabet $k + 1$.
– The initial number of objects: $a_0 + 1$.
– The number of rules: k.
– The sum of total length of rules:

Table 1. Resources of different order polynomial P systems. SWA, INO, NoR and SLR represent the size of the working alphabet, the initial number of objects, the number of rules and the sum of total length of rules, respectively. The symbol '–' means that the result is not obtained.

	Two-order	Three-order	...	k-order
SWA	3	4	...	k+1
INO	a_0+1	a_0+1	...	a_0+1
NoR	2	3	...	k
SLR	$3a_2 + a_1 + 5$	$7a_3 + 3a_2 + a_1 + 10$	...	$L_R(k)$

- Objects in the left-hand side: k.
- Objects **o** in the right-hand side: $\alpha_k^{(0)} + \alpha_k^{(1)} + \alpha_k^{(2)} + \cdots + \alpha_k^{(k-1)}$. That is, the total number of objects **o** in the rules is:

$$(2^k - 1) \cdot a_k + (2^{k-1} - 1) \cdot a_{k-1} + \cdots + (2^2 - 1) \cdot a_2 + (2^1 - 1) \cdot a_1$$

- Objects b_j $(1 \leq j \leq k)$ in the right-hand side:

$$\binom{j-1}{0} + \binom{j-1}{1} + \cdots + \binom{j-1}{j-1}$$

Thus, the total number of such kind of objects in the right hand side of the rules is:

$$\sum_{j=1}^{j=k} [\binom{j-1}{0} + \binom{j-1}{1} + \cdots + \binom{j-1}{j-1}] = \sum_{j=1}^{j=k} 2^{j-1} = 2^k - 1$$

Therefore, the sum of total length of rules is $L_R(k)$:

$$L_R(k) = k + (2^k - 1) \cdot a_k + (2^{k-1} - 1) \cdot a_{k-1} + \cdots + (2^2 - 1) \cdot a_2 + (2^1 - 1) \cdot a_1 + 2^k - 1$$

Hence, the total amount of descriptive computational resources is exponential in the k–order polynomial.

We summarize the amount of resources to build the designed P systems (of degree 1) as shown in Table 1, where SWA, INO, NoR and SLR represent the size of the working alphabet, the initial number of objects, the number of rules and the sum of total length of rules, respectively. In Table 1, the designed two-order polynomial P system (Two-order, for short) is $a_2 \cdot n^2 + a_1 \cdot n + a_0$; the designed three-order polynomial P system (Three-order, for short) is $a_3 \cdot n^3 + a_2 \cdot n^2 + a_1 \cdot n + a_0$; the k-order polynomial P system (k-order, for short) is $a_k \cdot n^k + ... + a_1 \cdot n + a_0$.

5 Conclusion

By analyzing the syntax and semantics of cell-like P systems, this study presented a reasoning method to design a k-order ($k \geq 2$) polynomial P system.

The significance of this study is to provide an alternative approach to automatically design a P system that can perform a given task, which is different from the previous work on this topic. In future works, we really hope to extend this method to design more variants of P systems for more computational tasks. Our ambitious aim is to find a way to design the minimal P system for a given task including definite tasks such as the computation of polynomials and indefinite tasks like practical applications such as membrane controllers for mobile robots.

Acknowledgment

The work of W. Yuan, G. Zhang, T. Wang and Z. Huang is supported by the National Natural Science Foundation of China (61170016, 61373047). The work of M.J. Pérez-Jiménez is supported by Project TIN2012-37434 of the Ministerio de Economía y Competitividad of Spain.

References

1. Gh. Păun. Computing with membranes, *Journal of Computer and System Sciences*, vol. 61, pp. 108–43 (2000) (first circulated at TUCS Research Report No 208, November 1998, http://www.tucs.fi)).
2. Gh. Păun, G. Rozenberg, A. Salomaa, eds. *The Oxford Handbook of Membrane Computing*, Oxford University Press, New York, 2010.
3. G. Zhang, J. Cheng, T. Wang, X. Wang, J. Zhu. *Membrane Computing: Theory and Applications*, Science Press, Beijing, 2015.
4. G. Zhang, M. Gheorghe, L. Pan, M.J. Pérez-Jiménez. Evolutionary membrane computing: a comprehensive survey and new results. *Information Sciences*, 279, 528–551 (2014).
5. G. Zhang, H. Rong, Z. Ou, M.J. Pérez-Jiménez, M. Gheorghe. Automatic Design of Deterministic and Non-Halting Membrane Systems by Tuning Syntactical Ingredients. *IEEE Transactions on Nanobioscience*, 13(3): 363–371 (2014).
6. G. Escuela, M.A. Gutiérrez-Naranjo. An application of genetic algorithms to membrane computing, in M.A. Martínez del Amor, Gh. Păun, I. Pérez Hurtado, A. Riscos (eds.) *Proceedings of the Eighth Brainstorming Week on Membrane Computing*, Seville, Spain, February 1-5, 2010, Report RGNC 01/2010, Fénix Editora, 2010, pp. 101-108.
7. X. Huang, G. Zhang, H. Rong, F. Ipate. Evolutionary design of a simple membrane system. *Lecture Notes in Computer Science (CMC2011)*, vol. 7184, 2012, pp. 203–14.
8. C. Tudose, R. Lefticaru, F. Ipate. Using genetic algorithms and model checking for P systems automatic design, *Studies in Computational Intelligence (NICSO2011)*, vol. 387, 2011, pp. 285–302.
9. Y. Chen, G. Zhang, T. Wang, X. Huang. Automatic design of a P system for basic arithmetic operations, *Chinese Journal of Electronics*, 23(2): 302–304 (2014).
10. Z. Ou, G. Zhang, T. Wang, and X. Huang. Automatic design of cell-like P systems through tuning membrane structures, initial objects and evolution rules, *International Journal of Unconventional Computing*, 9(5-6), 425–443 (2013).