Molecular Computing with Generalized Homogeneous P-Systems

Rudolf FREUND
Institut für Computersprachen
Technische Universität Wien
Karlsplatz 13
A-1040 Wien, Austria
email: rudi@logic.at
tel.: ++43 1 58801 18542

Franziska FREUND
De La Salle Schools
Strebersdorf
Anton Böck-Gasse 37
A-1215 Wien, Austria
email: ffreund@logic.at
tel.: ++43 1 58801 18542

Abstract. Recently P-systems were introduced by Gheorghe Păun as a new model for computations based on membrane structures. The basic variants of P-systems shown to have universal computational power only took account of the multiplicities of atomic objects, some other variants considered rewriting rules on strings. Using the membranes as a kind of filter for specific objects when transferring them into an inner compartment or out into the surrounding compartment turned out to be a very powerful mechanism in combination with suitable rules to be applied within the membranes in the model of generalized P-systems, GP-systems for short. GP-systems were shown to allow for the simulation of graph controlled grammars of arbitrary type based on productions working on single objects; moreover, various variants of GP-systems using splicing or cutting and recombination of strings were shown to have universal computational power, too. In this paper, we consider GP-systems with homogeneous membrane structures, GhP-systems for short, using splicing or cutting and recombination of string objects with specific markers at the ends of the strings that can be interpreted as electrical charges; the sum of these electrical charges determines the permeability of the membranes to the string objects. We show that GhP-systems using splicing or cutting and recombination have universal computational power; for GhP-systems using splicing the obtained results are optimal with respect to the underlying membrane structure and the uniformity of the permeability of the membranes. Moreover, a very restricted variant of such GhP-systems characterizes the (strictly) minimal linear languages.

1 Introduction

In the model of P-systems – as introduced by Gheorghe Păun in [10] – the most important feature is the membrane structure (for a chemical variant of this idea see [1]) consisting of membranes hierarchically embedded in the outermost *skin* membrane. Every membrane encloses a *region* possibly containing other membranes; the part delimited by the membrane labelled by k and its inner membranes is called *compartment* k. A region delimited by a membrane not only may enclose other membranes but also specific objects and operators, which in general are considered as multisets, as well as evolution rules, which in *generalized P-systems (GP-systems)* as introduced in [4] and [5] are evolution rules for the operators. In GP-systems, ground operators as well as transfer operators (simple rules of that kind are called travelling rules in [17]) are taken into account; these transfer operators transfer objects or operators (or even rules) either to the outer compartment or to an inner compartment delimited by a membrane of specific kind with also checking for some permitting and/or forbidding conditions on the objects to be transferred. In that way, the membranes act as a filter like in test tube systems (e.g., see [13] and [6]). In [5] it was shown how GP-systems with splicing or cutting and recombination rules can simulate test tube systems using the corresponding type of molecular rules. In contrast to the original definition of P-systems (e.g., see [2], [9]), in GP-systems no priority relations on the rules are used, and we do not enforce parallelism guarded by a universal clock.

Already in his first papers on membrane computing ([9]), Gheorghe Păun considered P-systems using the splicing operation on strings. P-systems based on splicing further were investigated in [15] and [16]. In all these variants, one main feature was that together with the application of a splicing rule the resulting string(s) actively could be moved out of a region or moved into an inner membrane. In [8] a specific model of generalized P-systems based on cutting and recombination with special homogeneous membrane structures were considered, the objects (assumed to be available in an unbounded number) being able to pass the membranes at a specific depth of the membrane structure depending on their electrical charges only.

In this paper we consider such GhP-systems (generalized homogeneous P-systems) based on splicing or cutting and recombination with a completely uniform permeability condition for all the membranes in the system, i.e., only objects with the absolute value of the difference of electrical charges being equal to 1 can pass a membrane in both directions. For GhP-systems based on splicing, the simplest non-trivial membrane structure is already sufficient for obtaining universal computational power. On the other hand, for GhP-systems based on cutting and recombination the optimality of the obtained results remains as an open problem.

Both the membrane structure as well as the operations used in GhP-systems based on splicing or cutting and recombination are motivated by nature, and moreover, the uniform permeability conditions for the membranes we are going to use in this paper look quite realistic. Yet despite this chemically/biologically motivated background, real implementations of GhP-systems - like implementations of other variants of P-systems - in the lab ("in vitro") remain a major challenge for the future.

In the following section we start with some preliminary notions from formal language theory and then give a general definition of a molecular system that also captures the notion of splicing and cutting/recombination systems of strings. In the third section we introduce the model of GhP-systems considered in this paper; as a first result, we show how a very restricted variant of GhP-systems characterizes the strictly minimal linear languages; moreover, we investigate the computational power of GhP-systems with the simplest membrane structure of depth zero. Our main result showing that GhP-systems using splicing or cutting/recombination have universal computational power is elaborated in the fourth section. A short summary and an outlook to future research topics and open problems conclude the paper.

2 Preliminary Definitions

For an alphabet V, by V^* we denote the free monoid generated by V under the operation of concatenation; the *empty string* is denoted by λ, and $|w|$ denotes the length of the string w, $w \in V^*$. $V^* \setminus \{\lambda\}$ is denoted by V^+; any subset of V^* (V^+) is called a *(λ-free) language*. $\mathbf{Z}$ denotes the set of integers; $\mathbf{N}$ denotes the set of non-negative integers.

A *minimal linear grammar* G is a quadruple $(\{S\}, V_T, P, S)$, where S is the only non-terminal symbol and the start symbol of the grammar, V_T is the terminal alphabet, and P is the set of linear productions of the forms $S \to w$ with $w \in V_T^*$ or $S \to uSw$ with $uw \in V_T^+$. If, moreover, $S \to \lambda$ is the only production in P of the form $S \to w$ with $w \in V_T^*$, then G is called a *strictly minimal linear grammar.*

A *molecular system* is a quadruple $\sigma = (B, B_T, P, A)$, where B and B_T are sets of *objects* and *terminal objects,* respectively, with $B_T \subseteq B$, P is a set of *productions,* and A is a set of axioms from B. A production p in P in general is a partial recursive relation $\subseteq B^k \times B^m$ for some $k, m \geq 1$, where we also demand that the domain of p is recursive (i.e., given $w \in B^k$ it is decidable if there exists some $v \in B^m$ with $(w,v) \in p$) and, moreover, that the range for every w is finite, i.e., for any $w \in B^k$, $card(\{v \in B \mid (w,v) \in p\}) < \infty$. For any two sets L and L' over B, we say that L' is computable from L by a production p if and only if for some $(w_1, ..., w_k) \in B^k$ and $(v_1, ..., v_m) \in B^m$ with $(w_1, ..., w_k, v_1, ..., v_m) \in p$ we have $\{w_1, ..., w_k\} \subseteq L$ and $L' = L \cup \{v_1, ..., v_m\}$; we also write $L \Longrightarrow_p L'$ and $L \Longrightarrow_\sigma L'$. A computation in σ is a sequence $L_0, ..., L_n$ such that $L_i \subseteq B$, $0 \leq i \leq n$, $n \geq 0$, as well as $L_i \Longrightarrow_\sigma L_{i+1}$, $1 \leq i \leq n$; in this case we also write $L_0 \Longrightarrow_\sigma^n L_n$, and moreover, we write $L_0 \Longrightarrow_\sigma^* L_n$ if $L_0 \Longrightarrow_\sigma^n L_n$ for some $n \geq 0$. The *language generated by* σ is

$$L(\sigma) = \{w \mid A \Longrightarrow_\sigma^* L,\ w \in L \cap B_T\}.$$

The special productions on string objects we shall consider in the following are the cutting and recombination operations as well as the splicing operation:

A *cutting/recombination scheme* (a *CR-scheme* for short) is a quadruple (V, M, C, R), where V is a finite alphabet; M is a finite set of *markers;* V and M are disjoint sets; C is a set of *cutting rules* of the form $u\#l\$m\#v$, where $u \in MV^* \cup V^*$, $v \in V^*M \cup V^*$, and $m, l \in M$, and $\#, \$$ are special symbols not in $V \cup M$; $R \subseteq M \times M$ is the recombination relation representing the *recombination rules.* Cutting and recombination rules are applied to objects from MV^*M. For $x, y, z \in MV^*M$ and a cutting rule $c = u\#l\$m\#v$ we define $x \Longrightarrow_c (y, z)$ if and only if for some $\alpha \in MV^*$ and $\beta \in V^*M$ we have $x = \alpha uv\beta$ and $y = \alpha ul$, $z = mv\beta$. For $x, y, z \in MV^*M$ and a recombination rule $r = (l, m)$ from R we define $(x, y) \Longrightarrow_r z$ if and only if for some $\alpha \in MV^*$ and $\beta \in V^*M$ we have $x = \alpha l$, $y = m\beta$, and $z = \alpha\beta$. For a CR-scheme $\sigma = (V, M, C, R)$ and any language $L \subseteq MV^*M$, $\sigma(L)$ then denotes the set of all objects obtained by applying one cutting or one recombination rule to objects from L. We also define $\sigma^0(L) = L$ and $\sigma^{i+1}(L) = \sigma(\sigma^i(L))$ for all $i \geq 0$, as well as $\sigma^{(0)}(L) = L$ and $\sigma^{(i+1)}(L) = \sigma^{(i)}(L) \cup \sigma\left(\sigma^{(i)}(L)\right)$ for all $i \geq 0$; moreover, we denote $\sigma^*(L) = \bigcup_{i=0}^{\infty} \sigma^{(i)}(L)$. An *extended CR-system* is a molecular system of type CR σ, $\sigma = (MV^*M, M_T V_T^* M_T, P, A)$, where $V_T \subseteq V$ is the set of terminal symbols, $M_T \subseteq M$ is the set of terminal markers, A is the set of axioms, P is the union of the relations (productions) defined by the cutting rules from C ($\subseteq (MV^*M) \times (MV^*M)^2$) and the recombination rules from R ($\subseteq (MV^*M)^2 \times (MV^*M)$), and (V, M, C, R) is the underlying CR-scheme.

Throughout this paper we shall restrict ourselves to markers that can be interpreted as electrical charges of ions, i.e., we shall write $[+k]$ and $[-k]$, $k \in \mathbf{N}$, for these special markers. In that sense, the recombination rules we use will be of the simple forms $([+k], [-k])$ and $([-k], [+k])$, $k \in \mathbf{N}$; the objects we are working with are of the form $[+k]\, w\, [-l]$ with $w \in V^*$. On such objects from $\mathbf{Z}'V^*\mathbf{Z}'$, where $\mathbf{Z}'$ is a finite subset of $\{[-k], [+k] \mid k \in \mathbf{N}\}$ (i.e., on linear objects with electrical charges at both ends) we can also define the splicing operation in the following way:

Let $\mathbf{Z}'$ be a finite subset of $\{[-k], [+k] \mid k \in \mathbf{N}\}$ and let V be a finite alphabet. An *extended splicing system (extended H system)* over $\mathbf{Z}'V^*\mathbf{Z}'$ is a molecular system of type H σ, $\sigma =$

$(\mathbf{Z}'V^*\mathbf{Z}', \mathbf{Z}''V_T^*\mathbf{Z}'', P, A)$, where $V_T \subseteq V$ is the set of terminal symbols, $\mathbf{Z}'' \subseteq \mathbf{Z}'$ is the set of terminal markers, A is the set of axioms, P is a set of splicing rules of the form $u_1\#u_2\$v_1\#v_2$, where $u_1, v_1 \in \mathbf{Z}'V^* \cup V^*$, $u_2, v_2 \in V^*\mathbf{Z}' \cup V^*$, and $\#, \$$ are special symbols not in $V \cup \mathbf{Z}'$; for $x, y, z \in \mathbf{Z}'V^*\mathbf{Z}'$ and a splicing rule $s = u_1\#u_2\$v_1\#v_2$ we define $(x, y) \Longrightarrow_s z$ if and only if for some $x_1, y_1 \in \mathbf{Z}'V^*$ and $x_2, y_2 \in V^*\mathbf{Z}'$ we have $x = x_1u_1u_2x_2$ and $y = y_1v_1v_2y_2$ as well as $z = x_1u_1v_2y_2$ (we omit the second result $y_1v_1u_2x_2$).

3 Generalized homogeneous P-systems (GhP-systems)

In this section we quite informally describe the model of GhP-systems discussed in this paper, especially the features not captured by the original model of P-systems as described in [2], [9], and [10]. In these papers, only the number of symbols is counted in the multiset sense, whereas in [14] at least the outputs are strings. In generalized P-systems (for the basic definition of GP-systems the reader is referred to [4] and [5]) the objects usually are strings or graphs, etc.

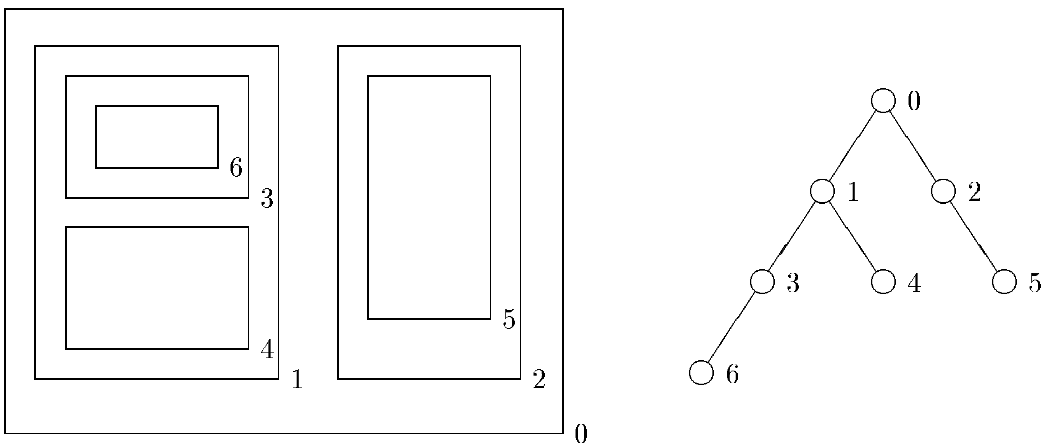

Figure 1: Membrane structure $[_0[_1[_3[_6]_6]_3[_4]_4]_1[_2[_5]_5]_2]_0$.

The basic ingredient of a (G(h))P-system is a *membrane structure* consisting of several membranes placed within one unique surrounding membrane, the so-called skin membrane. All the membranes can be labelled (in a one-to-one manner) by natural numbers; the outermost membrane (skin membrane) always is labelled by 0. In that way, a membrane structure can uniquely be described by a string of correctly matching parentheses, where each pair corresponds to a membrane. For example, the membrane structure depicted in Figure 1, which within the skin membrane contains two inner membranes labelled by 1 - containing membrane 3 (with membrane 6 inside) and membrane 4 - as well as by 2 (containing membrane 5) is described by $[_0[_1[_3[_6]_6]_3[_4]_4]_1[_2[_5]_5]_2]_0$. Figure 1 also shows that a membrane structure graphically can be represented by a Venn diagram, where two sets can either be disjoint or one set be the subset of the other one. In this representation, every membrane encloses a *region* possibly containing other membranes; the part delimited by the membrane labelled by k and its inner membranes is called *compartment* k in the following. The space outside the skin membrane is called *outer region*. Another obvious representation of a membrane structure is a tree as shown in Figure 1; in that sense the depth of a membrane structure can be defined as the depth of the corresponding tree; e.g., the depth of $[_0[_1[_3[_6]_6]_3[_4]_4]_1[_2[_5]_5]_2]_0$ is 3.

Informally, in [9] and [10] *P-systems* were defined as membrane structures containing multisets of objects in the compartments k as well as evolution rules for the objects. A priority relation on the evolution rules guarded the application of the evolution rules to the objects, which had to be affected in parallel (if possible according to the priority relation). The output was obtained in a designated compartment from a halting configuration (i.e., a configuration of the system where no rules can be applied any more).

A *generalized homogeneous P-system (GhP-system) of molecular type* X is a construct γ, $\gamma = (B, B_T, P, A, \mu, I, in, out)$, where

- (B, B_T, P, A) is a molecular system of type X (we shall consider molecular systems of type CR and H in the following);
- μ is a membrane structure (with the membranes labelled by natural numbers $0, ..., n$);
- $I = (I_0, ..., I_n)$, where I_k is the initial contents of compartment k containing a set of objects from A as well as a set of rules from P;
- in for each $j \in \{0, 1, ..., n\}$ specifies a condition an object must fulfill in order to be able to pass into the inner compartment of a membrane k, $k \in \{1, ..., n\}$, in the region enclosed by membrane j;
- out specifies a condition an object must fulfill in order to be able to pass a membrane j, $j \in \{0, 1, ..., n\}$, into its surrounding compartment.

A *computation* in γ starts with the initial configuration with I_k being the contents of compartment k. In contrast to the original definition of P-systems, the objects in the compartments are not treated in the multiset sense; instead we assume all objects occuring in a compartment to be available in an unbounded number. A transition from one configuration to another one is performed by applying a rule (from P) in I_k to objects in compartment k or by moving an element out of a compartment k or into a compartment k according to the conditions given by out and in. The language generated by γ is the set of all terminal objects $w \in B_T$ obtained in the outer region by some computation in γ.

In test tube systems (e.g., see [6]), the contents of the tubes is redistributed to all other tubes according to specific input or output filters; the operations in the tubes are based on molecular systems of type H or CR. In [6] we showed that test tube systems with only two test tubes using splicing or cutting and recombination rules in the tubes and filters of a special type between the tubes have the computational power of arbitrary grammars and Turing machines, respectively. According to the definition given above, in the GhP-systems of type CR used in [8], we only allowed one unique out-condition, whereas the in-conditions specified by in depended on the region the membrane to be passed lay in, where the in-conditions of regions at the same level of the membrane structure coincided; GhP-systems of type CR with a membrane structure of the form $[_0[_1[_{n+1}]_{n+1}]_1 ... [_n[_{2n}]_{2n}]_n]_0$ of depth two were shown to have universal computational power in [8].

In this paper we will use a completely uniform permeability condition for the membranes, i.e., only objects with the absolute value of the sum of electrical charges being equal to 1, can pass the membranes in both directions; hence in the following, we shall not specify this uniform static in- and out-conditions any more. Although using this special permeability condition for the membranes, we can improve the results for GhP-systems of type CR shown in [8], i.e., we can reduce the complexity of the membrane structure to depth one, and for GhP-systems of type H we can establish even optimal results, i.e., even with the simplest non-trivial membrane structure $[_0[_1]_1]_0$ we obtain universal computational power (compare with the results proved in [5], [15], and [16]).

To illustrate the model of GhP-systems, we give an example of a GhP-system of type CR generating the (non-regular) language $\{a^n b^n \mid n \geq 0\}$:

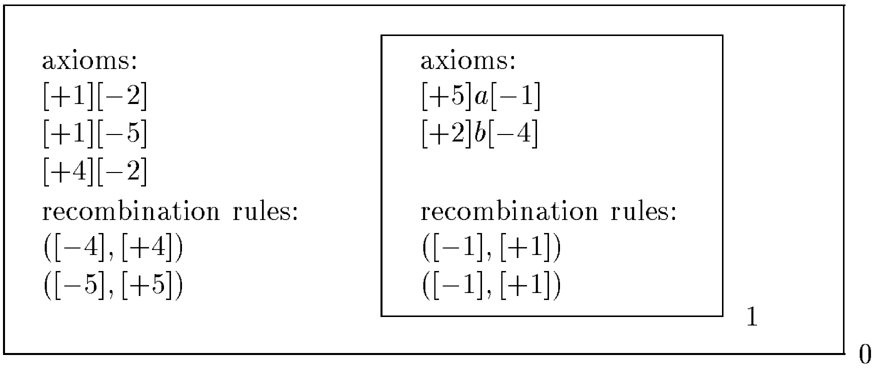

Figure 2: GhP-system generating $\{a^n b^n \mid n \geq 0\}$.

Example 1. The main ingredients of a GhP-system of type CR generating the language $\{a^n b^n \mid n \geq 0\}$ are depicted in Figure 2. In the skin membrane we start with (terminal) objects of the form $[+1]a^n b^n[-2]$ for some $n \geq 0$; these objects can enter compartment 1, where one application of each of the recombination rules $([-1],[+1])$ and $([-2],[+2])$ with the corresponding axioms yields $[+5]a^{n+1}b^{n+1}[-4]$, which object can pass back to compartment 0, where the application of the recombination rules $([-4],[+4])$ and $([-4],[+4])$ yields the terminal object $[+1]a^{n+1}b^{n+1}[-2]$. □

Replacing the recombination rules $([-k],[+k])$, $k \in \{1,2,4,5\}$, by the corresponding special splicing rules $\#[-k]\$[+k]\#$ yields a GhP-system of type H generating $\{a^n b^n \mid n \geq 0\}$.

In a similar way as it was shown for the language $\{a^n b^n \mid n \geq 0\}$ in Example 1, all strictly minimal linear languages can be generated by GhP-system of type CR using only recombination rules in a membrane structure of depth 1:

Lemma 2. *Any strictly minimal linear language generated by a strictly minimal linear grammar G, $G = (\{S\}, V_T, P, S)$, can be generated by a GhP-system of type CR using only recombination rules within the membrane structure $[_0[_1]_1 \ldots [_n]_n]_0$, where n is the number of linear productions of the form $S \rightarrow uSw$ with $uw \in V_T^+$ in P.*

Proof. The main parts of the GhP-system are depicted in Figure 3. The terminal objects $w \in L(G)$ can leave compartment 0 in the form $[+1]w[-2]$. □

Without further proof we mention that GhP-systems of type CR of the very restricted form as constructed in the preceding lemma exactly characterize the strictly minimal linear languages. Moreover, as already mentioned above, the recombination rule $([-k],[+k])$ can be simulated by the splicing rule $\#[-k]\$[+k]\#$, hence, in that way we also obtain a characterization of strictly minimal linear languages by GhP-systems of type H.

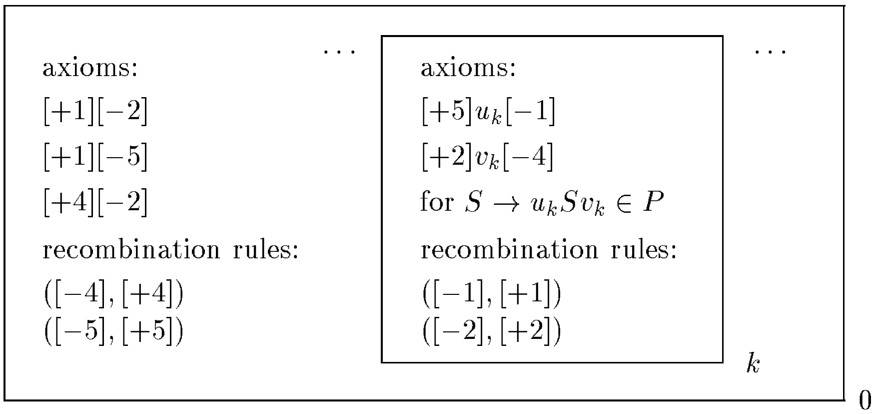

Figure 3: GhP-system of type CR generating $L(G)$, where $G = (\{S\}, V_T, P, S)$ is a strictly minimal linear grammar.

To the end of this section, we investigate the computational power of GhP-systems with the simplest membrane structure $[_0]_0$.

Lemma 3. *The family of regular languages can be characterized by GhP-systems of type H/CR with the simplest membrane structure* $[_0]_0$.

Proof. Extended H-systems as well as extended CR-systems exactly characterize the family of regular languages (see [13] and [18]). Hence, if we consider a GhP-system of type H/CR with the simplest membrane structure $[_0]_0$, then the resulting language in compartment 0 is regular. As the family of regular languages is closed under intersection, the language obtained in the outer region as the intersection of this language in compartment 0 and a regular language of the form $\mathbf{Z}''V_T^*\mathbf{Z}''$ with $\mathbf{Z}''$ being a finite subset of $\{[-k], [+k] \mid k \in \mathbf{N}\}$ again is regular.

On the other hand, let G, $G = (V_N, V_T, P, S)$, be a regular grammar with productions in P of the form $B \to cD$ or $B \to c$, $B, D \in V_N$, $c \in V_T$:

$L(G)$ is generated by the GhP-system of type H γ :

$$
\begin{aligned}
\gamma &= (B, B_T, R, A, [_0]_0, (A \cup R)) \\
B &= \mathbf{Z}'V^*\mathbf{Z}' \\
\mathbf{Z}' &= \{[+k], [-k] \mid 1 \le k \le 4\} \\
V &= V_N \cup V_T \\
B_T &= \mathbf{Z}''V_T^*\mathbf{Z}'' \\
\mathbf{Z}'' &= \{[+k], [-k] \mid 1 \le k \le 2\} \\
R &= R_1 \cup R_2 \\
R_1 &= \{u\#B\,[-4]\,\$\,[+4]\,\#cD\,[-4] \mid B \to cD \in P,\ u \in W\} \\
R_2 &= \{u\#B\,[-4]\,\$\,[+4]\,\#c\,[-1] \mid B \to c \in P,\ u \in W\} \\
W &= \{[+1]\} \cup \{[+1]\}\,V_T \cup V_T^2 \\
A &= \{[+1]\,S\,[-4]\} \cup A_1 \cup A_2 \\
A_1 &= \{[+4]\,cD\,[-4] \mid B \to cD \in P\} \\
A_2 &= \{[+4]\,c\,[-2] \mid B \to c \in P\}
\end{aligned}
$$

The application of a production $B \to cD \in P$ is simulated by using a suitable splicing rule of the form $u\#B\,[-4]\,\$\,[+4]\,\#cD\,[-4]$ together with the axiom $[+4]\,cD\,[-4]$, the application of a terminal production $B \to c \in P$ is simulated by using a suitable splicing rule of the form $u\#B\,[-4]\,\$\,[+4]\,\#c\,[-2]$ together with the axiom $[+4]\,c\,[-2]$, thus finally yielding a terminal object $[+1]\,w\,[-2]$ for some $w \in V_T^+$.

A GhP-system of type CR γ, $\gamma = (B, B_T, R, A, [_0]_0, (A \cup R))$, generating $L(G)$ is more complicated, because in this case we have to store the information about the non-terminal symbol at the end of the sentential form in the marker on the right-hand side; on the other hand, the reader should observe that we only need recombination rules, but no cutting rules:

Let $V_N = \{X_{2i} \mid 2 \leq i \leq m\}$ and $S = X_4$. Then we take $B = \mathbf{Z}'V^*\mathbf{Z}'$, $\mathbf{Z}' = \{[+k], [-k] \mid 1 \leq k \leq 2m\}$, and V and B_T as above; moreover, we now take the following recombination rules and axioms:

$$\begin{aligned}
R &= \{([-2i], [+2i]) \mid 2 \leq i \leq m\} \\
A &= \{[+1]\,[-4]\} \cup A_1 \cup A_2 \\
A_1 &= \{[+2i]\,c\,[-2j] \mid X_{2i} \to cX_{2j} \in P\} \\
A_2 &= \{[+2i]\,c\,[-2] \mid X_{2i} \to c \in P\}
\end{aligned}$$

It is easy to see that a terminal object $[+1]\,w\,[-2]$ for some $w \in V_T^+$ is generated in compartment 0 if and only if $w \in L(G)$. □

In the proofs of Lemma 3 we could also take $B_T = B$, because the filtering out of the terminal objects could be achieved by the passing condition for the skin membrane 0 only.

In contrast to the rather obvious result established for GhP-systems of type H in Lemma 3, where we only need a limited number of markers, the number of markers is not bounded in the GhP-systems of type CR and depends on the number of non-terminal symbols in the underlying regular grammar, which fact may give rise to an infinite non-collapsing hierachy with respect to the number of markers.

4 The computational power of GhP-systems of type H and CR

The main result we prove in the following establishes the universal computational power of GhP-systems. First we improve the result established for GhP-systems of type CR in [8]:

Theorem 4. *Any recursively enumerable language L can be generated by a GhP-system γ of type CR with a fixed number of markers in the membrane structure* $[_0[_1]_1 ... [_n]_n]_0$.

Proof. The proof idea to simulate the productions of a grammar generating L like in Post systems in normal form ("rotate-and-simulate") has already been used in many papers on splicing systems and cutting/recombination systems (see [13] and [6]). Moreover, instead of a grammar G' generating L, we consider a grammar G, $G = (V_N \cup \{B\}, V_T \cup \{d\}, P, S)$, generating the language $L\{d\}$, where the end marker d, $d \notin V_N \cup \{B\} \cup V_T$, in any derivation of a word $w'd$, for $w' \in L$, is generated exactly in the last step of this derivation in G and for each symbol $X \in V_N \cup V_T \cup \{B\}$ the production $X \to X$ is in P. A string $w \in (V_N \cup V_T)^*$ appearing in a derivation of such a grammar G generating $L\{d\}$, is represented by its rotated versions $[+3]\,Xw_2Bw_1Y\,[-4]$, where $w = w_1w_2$ and B is a special symbol indicating the beginning of the string within the rotated versions and X, Y are special symbols marking the ends of the strings. Final strings first appear in the form $[+7]\,XdBw'Y\,[-6]$, where w' is the final result from L which we want to get, and finally leave compartment 0 in the form $[+1]\,w'\,[-2]$.

The GhP-system γ we construct, for each production $p_k : \alpha_k \to \beta_k$, $1 \leq k \leq n$, where without loss of generality we assume $1 \leq |\alpha_k| \leq 2$, $0 \leq |\beta_k| \leq 2$, and $||\alpha_k| - |\beta_k|| \leq 1$, contains compartment k :

For simulating p_k in compartment k, $1 \leq k \leq n$, we use the *cutting rules*

$u\#\,[-1]\,\$\,[+1]\,\#\alpha_k Y\,[-6]$, $u \in V_N \cup V_T \cup \{B, X\}$, and

$[+3]\,X\#\,[-5]\,\$\,[+5]\,\#v$, $v \in V_N \cup V_T \cup \{B\}$,

as well as the *recombination rules*

$([-1]\,,[+1])$ and

$([-5]\,,[+5])$

together wih the axioms

$[+1]\,Y\,[-6]$ and

$[+7]\,X\beta_k\,[-1]$.

Thus we obtain $[+7]\,X\beta_k wY\,[-6]$ from $[+3]\,Xw\alpha_k Y\,[-4]$.

In compartment 0 we start with the axiom $[+3]XBSY[-4]$; the axioms $[+3]\,[-7]$ and $[+6]\,[-4]$ together with the recombination rules $([-6]\,,[+6])$ and $([-7]\,,[+7])$ allow us to obtain $[+3]\,X\beta_k wY\,[-4]$ from $[+7]\,X\beta_k wY\,[-6]$ thus finishing the simulation of p_k. An object of the form $[+7]\,XvY\,[-6]$ could also pass immediately into each of the compartments k, $1 \leq k \leq n$, but there it could not be processed, and therefore the only possibility would be to leave compartment k unchanged again.

Final strings first appearing in the form $[+7]\,XdBw'Y\,[-6]$, $w' \in L$, can leave compartment 0 in the form $[+1]\,w'\,[-2]$ after the application of suitable cutting rules $[+7]\,XdB\#\,[-1]\,\$\,[+1]\,\#a$, $a \in V_T \cup \{Y\}$, and $b\#\,[-2]\,\$\,[+2]\,\#Y\,[-6]$, $b \in V_T \cup \{[+1]\}$. If a cutting rule $b\#\,[-2]\,\$\,[+2]\,\#Y\,[-6]$ is applied too early, we obtain an object $[+7]\,Xau\,[-2]$ with $a \neq d$, which object cannot be processed any more. A terminal object $[+1]\,w'\,[-2]$ not only can leave compartment 0, but also enter any other compartment k, $1 \leq k \leq n$, yet recombining this object with another object there yields objects of the form $[+7]\,X\beta_k ww'\,[-2]$, $[+5]\,Xvw'\,[-2]$ - these objects cannot be processed any more - or $[+3]\,Xvw'\,[-2]$, which object can pass to every component, but it is not terminal and therefore cannot leave compartment 0, but it also cannot be processed any more.

$$
\begin{array}{rcl}
\gamma & = & (B, B_T, R, A, [_0[_1]_1 ... [_n]_n]_0, (A_0 \cup R_0, A_1 \cup R_1, ..., A_n \cup R_n)) \\
B & = & \mathbf{Z}' V^* \mathbf{Z}' \\
\mathbf{Z}' & = & \{[+k]\,,[-k] \mid 1 \leq k \leq 7\} \\
V & = & V_N \cup V_T \cup \{d, B, X\} \cup \\
B_T & = & \mathbf{Z}'' V_T^* \mathbf{Z}'' \\
\mathbf{Z}'' & = & \{[+k]\,,[-k] \mid 1 \leq k \leq 2\} \\
R & = & R_0 \cup R_1 \cup ... R_n \\
A & = & A_0 \cup A_1 \cup ... A_n
\end{array}
$$

The sets of axioms are

$$
\begin{array}{rcl}
A_0 & = & \{[+3]XBSY[-4], [+3]\,[-7]\,,[+6]\,[-4]\} \text{ and} \\
A_k & = & \{[+1]\,Y\,[-6]\} \cup \{[+7]\,X\beta_k\,[-1] \mid 1 \leq k \leq n\} \text{ for } 1 \leq k \leq n.
\end{array}
$$

The sets of rules $R_0, ..., R_n$ are specified by the following sets of cutting rules and recombination rules:

$$R_0 = \{([-6],[+6]),([-7],[+7])\} \cup \{[+7]\,XdB\#\,[-1]\,\$\,[+1]\,\#a \mid a \in V_T \cup \{Y\}\} \cup \{b\#\,[-2]\,\$\,[+2]\,\#Y\,[-6] \mid b \in V_T \cup \{[+1]\}\}$$

$$R_k = \{([-1],[+1]),([-5],[+5])\} \text{ for } 1 \leq k \leq n.$$

The remaining technical details of the proof are left to the reader. □

The complexity of the GhP-system of type CR constructed in the preceding proof depends on the number of productions (and symbols) in the underlying grammar, which may give rise to a non-collapsing hierarchy with respect to the number of inner membranes. In contrast, for GhP-systems of type H we obtain an optimal result with respect to the membrane structure (compare with Lemma 3).

Theorem 5. *Any recursively enumerable language L can be generated by a GhP-system γ of type H with a fixed number of markers in the simplest non-trivial membrane structure* $[_0[_1]_1]_0$.

Proof. The main proof idea is the same as in the preceding proof, yet we also take advantage of the idea used in some proofs in [15] how to check the correspondence of the symbols marking the left and the right end of an intermediate string.

For simulating the production $p_k : \alpha_k \to \beta_k$, $1 \leq k \leq n$, in compartment 1, we start with applying

$u\#\alpha_k Y\,[-3]\,\$\,[+3]\,\#Y_k\,[-6]$, $u \in V_N \cup V_T \cup \{B\}$, and

$[+7]\,X_k\beta_k\#\,[-2]\,\$\,[+2]\,X\#v$, $v \in V_N \cup V_T \cup \{B, X\}$,

using the axioms $[+3]\,Y_k\,[-6]$ and $[+7]\,X_k\beta_k\,[-2]$.

In compartment 0, where we start with the axiom $[+2]XBSY[-3]$, the indices of the variables X and Y are decremented by applying

$[+4]\,X_{i-1}\#\,[-7]\,\$\,[+7]\,X_i\#v$, $v \in V_N \cup V_T \cup \{B, d\}$, $1 \leq i \leq n$, and

$u\#Y_j\,[-6]\,\$\,[+6]\,\#Y_{j-1}\,[-5]$, $u \in V_N \cup V_T \cup \{B\}$, $1 \leq j \leq n$,

using the axioms $[+4]\,X_{i-1}\,[-7]$ and $[+6]\,Y_{j-1}\,[-5]$.

Objects of the form $[+4]\,X_iwY_j\,[-5]$ can pass into compartment 1, where we apply

$u\#Y_j\,[-5]\,\$\,[+5]\,\#Y_j\,[-6]$, $u \in V_N \cup V_T \cup \{B\}$, and

$[+7]\,X_i\#\,[-5]\,\$\,[+5]\,X_i\#v$, $v \in V_N \cup V_T \cup \{B, d\}$,

using the axioms $[+5]\,Y_j\,[-6]$ and $[+7]\,X_i\,[-5]$, for $0 \leq i, j \leq n-1$, in order to obtain $[+7]\,X_iwY_j\,[-6]$.

If in the first steps in compartment 1 the correctly matching splicing rules have been applied, we finally reach compartment 0 with an object of the form $[+7]\,X_0wY_0\,[-6]$, which is the only case that allows us to regain an object of the form $[+2]\,XwY\,[-3]$ by applying the splicing rules

$[+2]\,X\#\,[-7]\,\$\,[+7]\,X_0\#v$, $v \in V_N \cup V_T \cup \{B\}$, and

$u\#Y_0\,[-6]\,\$\,[+6]\,\#Y\,[-3]$, $u \in V_N \cup V_T \cup \{B\}$,

using the axioms $[+2]\,X\,[-7]$ and $[+6]\,Y\,[-3]$.

From terminating objects of the form $[+7]\,X_0dBw'Y_0\,[-6]$ with $w' \in L$ we obtain the terminal objects $[+1]\,w'\,[-2]$ by applying the splicing rules

$[+1]\,\#X\,[-7]\,\$\,[+7]\,X_0dB\#a$, $a \in V_T \cup \{Y_0\}$, and

$b\#Y_0\,[-6]\,\$\,[+6]\,Y\#\,[-2]$, $b \in V_T \cup \{[+1]\}$,

using the axioms $[+1]\,X\,[-7]$ and $[+6]\,Y\,[-2]$.

From the list of axioms and splicing rules in the compartments 0 and 1 as described above the formal description of the GhP-system γ of type H can easily be completed; as some of the axioms, i.e., $[+6]\,Y_j\,[-5]$ and $[+5]\,Y_j\,[-6]$, for $0 \le j \le n-1$, can travel between compartment 0 and compartment 1, without violating the correct functioning of γ, we can assign each axiom listed above to both compartments; on the other hand, the power of the GhP-system lies in the correct assignment of the splicing rules:

$$
\begin{aligned}
\gamma &= (B, B_T, R, A, [_0]_0, (A \cup R_0, A \cup R_1)) \\
B &= \mathbf{Z}' V^* \mathbf{Z}' \\
\mathbf{Z}' &= \{[+k]\,,[-k] \mid 1 \le k \le 7\} \\
V &= V_N \cup V_T \cup \{d, B, X\} \cup \\
B_T &= \mathbf{Z}'' V_T^* \mathbf{Z}'' \\
\mathbf{Z}'' &= \{[+k]\,,[-k] \mid 1 \le k \le 2\} \\
R &= R_0 \cup R_1
\end{aligned}
$$

The complete set of axioms is

$$
\begin{aligned}
A = {} & \{[+3]\,Y_k\,[-6]\,,[+7]\,X_k\beta_k\,[-2] \mid 1 \le k \le n\} \cup \\
& \{[+4]\,X_k\,[-7]\,,[+6]\,Y_k\,[-5]\,,[+5]\,Y_k\,[-6]\,,[+7]\,X_k\,[-5] \mid 0 \le k \le n-1\} \cup \\
& \{[+2]XBSY[-3], [+2]\,X\,[-7]\,,[+6]\,Y\,[-3]\,,[+1]\,X\,[-7]\,,[+6]\,Y\,[-2]\}\,,
\end{aligned}
$$

the splicing rules are collected in R_0 and R_1 as follows:

$$
\begin{aligned}
R_0 = {} & \{[+4]\,X_{i-1}\#\,[-7]\,\$\,[+7]\,X_i\#v \mid v \in V_N \cup V_T \cup \{B, d\}\,, 1 \le j \le n\} \cup \\
& \{u\#Y_j\,[-6]\,\$\,[+6]\,\#Y_{j-1}\,[-5] \mid u \in V_N \cup V_T \cup \{B\}\,, 1 \le j \le n\} \cup \\
& \{[+2]\,X\#\,[-7]\,\$\,[+7]\,X_0\#v \mid v \in V_N \cup V_T \cup \{B\}\} \cup \\
& \{u\#Y_0\,[-6]\,\$\,[+6]\,\#Y\,[-3] \mid u \in V_N \cup V_T \cup \{B\}\} \cup \\
& \{[+1]\,\#X\,[-7]\,\$\,[+7]\,X_0dB\#a \mid a \in V_T \cup \{Y_0\}\} \cup \\
& \{b\#Y_0\,[-6]\,\$\,[+6]\,Y\#\,[-2] \mid b \in V_T \cup \{[+1]\}\} \\
R_1 = {} & \{u\#\alpha_k Y\,[-3]\,\$\,[+3]\,\#Y_k\,[-6] \mid u \in V_N \cup V_T \cup \{B\}\,, 1 \le k \le n\} \cup \\
& \{[+7]\,X_k\beta_k\#\,[-2]\,\$\,[+2]\,X\#v \mid v \in V_N \cup V_T \cup \{B, X\}\,, 1 \le k \le n\} \cup \\
& \{u\#Y_k\,[-5]\,\$\,[+5]\,\#Y_k\,[-6] \mid u \in V_N \cup V_T \cup \{B\}\,, 0 \le k \le n-1\} \cup \\
& \{[+7]\,X_k\#\,[-5]\,\$\,[+5]\,X_k\#v \mid v \in V_N \cup V_T \cup \{B, d\}\,, 0 \le k \le n-1\}
\end{aligned}
$$

According to the definitions and explanations given above, the reader may easily verify that $L(\gamma) = L$, which completes the proof. □

The variables X_k and Y_k used in the preceding proof can be encoded in the markers; hence, allowing an unbounded number of markers, the proof idea elaborated above can also be used for showing the following result, i.e., there is a trade-off between the number of membranes and the number of markers to be used in GhP-systems of type CR:

Corollary 6. *Any recursively enumerable language L can be generated by a GhP-system γ of type CR in the simplest non-trivial membrane structure $[_0[_1]_1]_0$.*

Obviously, we can encode arbitrary additional data u in the axiom, i.e., we can take $[+k]XBSuY[-k-1]$ instead of $[+k]XBSY[-k-1]$ in compartment 0; moreover, starting with these axioms, the grammars can be considered as computing devices, the terminal strings $[+1]\,w'\,[-2]$ representing the output w' computed from the input u. Hence, from the preceding theorems we also obtain the following result:

Corollary 7. *Any partial recursive function can be computed by a GhP-system of type CR and by a GhP-system of type H, respectively, using a fixed number of markers only.*

The complexity of the GhP-systems of type CR and H, respectively, in Corollary 7 is the same as of the corresponding GhP-systems constructed in the proofs of Theorems 4 and 5, i.e. for GhP-systems of type H the result is optimal with respect to the membrane structure and a fixed number of markers.

5 Conclusion

As it was already pointed out in [10], the idea of membrane structures offers a nearly unlimited variety of variants. In this paper we considered homogeneous (static) membranic structures with splicing or cutting/recombination rules inside and with the uniform permeability of the membranes to neutral objects. The formal definition of molecular systems would also allow us to consider other objects than strings, e.g., graphs and the corresponding cutting and recombination rules (see [7]). A thorough investigation of the generative power of such variants of GhP-systems and their complexity for solving special problems remains for future research.

Further interesting features for P-systems can be found, for example, in [3], where the communication of objects is controlled by the concentration of the objects, and in [11], where the communication through membranes depends on the electrical charges of the atomic objects and the membranes. In [12], Gheorghe Păun gives an overview on the actual bibliography of P-systems and discusses a list of interesting problems related with membrane computing.

We finish with some open problems arising from the results discussed in this paper:

- Consider the simple GhP-systems of type CR using only recombination rules especially as in Lemmas 2 and 3:

  1. We possibly may have the chance to obtain an infinite non-collapsing hierarchy (compare with problem “m” in [12]) with respect to n in the membrane structure $[_0[_1]_1 \ldots [_n]_n]_0$. The family of languages $\{ww^r \mid w \in \Sigma^*\}$ (where w^r denotes the mirror image of w) may be a candidate to constitute such a hierarchy with respect to the number of symbols in Σ.
  2. For the simplest membrane structure $[_0]_0$ we may obtain an infinite non-collapsing hierarchy with respect to the number of markers $[+k]\,,[-k]$.

- Can the complexity (the number of membranes) of the GhP-systems of type CR constructed in the proof of Theorem 4 be reduced without increasing the number of markers?

- Can the the number of markers of the GhP-systems of type H constructed in the proof of Theorem 5 be reduced?

Acknowledgements

We gratefully acknowledge all the fruitful and interesting discussions with Gheorghe Păun concerning his brilliant ideas for various models of P-systems.

References

[1] G. Berry and G. Boudol, *The chemical abstract machine,* Theoretical Computer Science 96 (1992), pp. 217-248.

[2] J. Dassow and Gh. Păun, *On the power of membrane computing,* JUCS **5** (2) (1999), pp. 33-49.

[3] J. Dassow and Gh. Păun, *Concentration controlled P systems,* submitted, 1999.

[4] R. Freund, *Generalized P-systems,* FCT'99, Iasi, Romania, September 1999.

[5] R. Freund, *Generalized P-systems with splicing and cutting/recombination,* WFLA'99, Iasi, Romania, September 1999.

[6] R. Freund and F. Freund, *Test tube systems: When two tubes are enough,* DLT'99, Aachen, Germany, July 1999.

[7] R. Freund and F. Freund, *Cutting and recombination of graphs,* AFL'99, Hungary, August 1999.

[8] R. Freund and F. Freund, *Generalized homegeneous P-systems,* Research Report TU Wien, Austria, 2000.

[9] Gh. Păun, *Computing with membranes,* TUCS Research Report No. 208 (Nov. 1998).

[10] Gh. Păun, *Computing with membranes: an introduction,* Bulletin EATCS **67** (Febr. 1999), pp. 139-152.

[11] Gh. Păun, *Computing with membranes - a variant: P systems with polarized membranes,* Auckland University, CDMTCS Report No. 098, 1999.

[12] Gh. Păun, *Computing with membranes (P systems): Twenty-six research topics,* Research Report, 2000.

[13] Gh. Păun, G. Rozenberg, and A. Salomaa, *DNA Computing: New Computing Paradigms* (Springer-Verlag, Berlin, 1998).

[14] Gh. Păun, G. Rozenberg, and A. Salomaa, *Membrane computing with external output,* TUCS Research Report No. 218 (Dec. 1998).

[15] Gh. Păun and T. Yokomori, *Membrane computing based on splicing,* Preliminary Proc. Fitfth Intern. Meeting on DNA Based Computers (E. Winfree and D. Gifford, eds.), MIT, June 1999, pp. 213-227.

[16] Gh. Păun and T. Yokomori, *Simulating H systems by P systems,* JUCS **6** (2) (2000), pp. 178-193.

[17] I. Petre, *A normal form for P-systems,* Bulletin EATCS **67** (Febr. 1999), pp. 165-172.

[18] D. Pixton, *Splicing in abstract families of languages,* Technical Report SUNY Univ. at Binghamton, New York, (1997).