# On two variants of splicing super-cell systems[1]

Pierluigi Frisco
LIACS, Leiden University,
Niels Bohwerg 1, 2333 CA Leiden, The Netherlands
e-mail: pier@liacs.nl

**Abstract.** New computability models, called super-cell systems or P systems, based on the evolution of objects in a membrane structure, were recently introduced. The seminal paper of Gheorghe Păun describes three ways to look at them: transition, rewriting and splicing super-cell systems having different properties.
Here we investigate two variants of splicing super-cell systems improving results concerning their generative capability. This is obtained with a variant of the "rotate-and-simulate" technique classical in H systems area.

## 1 Introduction

Super-cell systems (also called P systems) were recently introduced in [5] as distributed parallel computing models.

In the seminal paper the author considers systems based on a hierarchical arranged, finite *cell-structure* consisting of several cell-membranes embedded in a main membrane called *skin*. The membranes delimit *regions* where *objects*, elements of a finite set or alphabet, and *evolution rules* can be placed.

The objects evolve according to given *evolution rules* associated with a region; priorities can be associated to evolution rules. They contain symbols as $a_{here}, a_{out}$ or $a_{in_i}$ where $a$ is an object. The meaning of the subscripts is: *here* indicates that the object remains in the membranes in which it was produced; *out* means that the object in sent out of the membranes in which it was produced; $in_i$ means that the object is sent to membrane $i$ if it is reachable from the region where the rule is applied, if not the rule is not applied.

The objects can evolve independently or in cooperation with other objects present in the region in which they are. An evolution rule can destroy the membrane in which it is. In this case all the objects of the destroyed membrane pass to the immediately superior one and they evolve according to this one's evolution rules. The rules of the dissolved cell are lost. The skin membrane cannot be dissolved.

Such a system evolves in parallel: at each step all objects which can evolve do it. A *computation* starts from an initial configuration of a system, defined by a cell-structure with objects and evolution rules in each cell, and terminates when no further rule can be applied.

It is possible to assign a result to a computation in two ways: considering the multiplicity of objects present in a designed membrane in a halting configuration,

[1]This is the extended and corrected version of P. Frisco "Membrane computing based on splicing: improvements", present in pre-proceedings of Workshop on Multiset Processing, 21 - 25 August 2000, Curtea de Arges, Rumania

or concatenating the symbols leaving the system in the order they are sent out of the skin membrane.

In [5] the author examines three ways to look at P systems: transition, rewriting and splicing super-cell systems. Starting from these several variants were considered: [6] gives a survey; in [7] polarized membranes and "electrical charges" assigned to objects are considered; in [10] rules with $a_{in}$ (indicating that an object passes to any of the adjacent lower membranes non-deterministically chosen) and other types of structures (planar maps described by asymmetric graphs) are introduced; in [11] variants of splicing P systems with or without planar map are investigated. In most of the cases the characterization of recursively enumerable (RE) number relations or representation of permutation closures of RE languages are obtained.

We focused our attention on some of the systems introduced and studied in [11]. The objects of our investigations are P systems using string-object evolving by splicing with non-deterministic way of communicating and P system using string-object evolving by splicing working on planar maps described by asymmetric graphs. The characterization of RE languages is improved reducing the degree and the depth of the systems. One minimal result concerning the number of membrane used is obtained.

## 2 Splicing and P systems

The operation of splicing as a formal model of DNA recombination with the presence of restriction enzymes and ligases was introduced in [2]. Now we give definitions strictly related with our work; more general information may be found in [9].

Consider an alphabet $V$ and two special symbols, $\#$ and $\$$ not in $V$. With $V^*$ we indicate the free monoid generated by the alphabet $V$ under the operation of concatenation; $\lambda$ indicates the empty string; the length of $x \in V^*$ (the number of symbols of $V$ present in $x$) is indicated with $|x|$.

A *splicing rule* is a string of the form $r = u_1\#u_2\$u_3\#u_4$, where $u_1, u_2, u_3, u_4 \in V^*$. For such a splicing rule $r$ and strings $x, y, z, w \in V^*$ we write:

$$\begin{aligned}(x, y) \vdash_r (z, w) \quad \text{iff} \quad & x = x_1u_1u_2x_2, y = y_1u_3u_4y_2, \\ & z = x_1u_1u_4y_2, w = y_1u_3u_2x_2, \\ & \text{for some } x_1, x_2, y_1, y_2 \in V^*.\end{aligned} \tag{1}$$

What just defined is called *2-splicing* as two strings, $z$ and $w$, are obtained as output. For a 2-splicing we call $z$ and $w$ the first and the second output string respectively.

In (1) it is also possible to consider only $z$ as output. In this case the operation is called *1-splicing*.

Considering a rule $r$ as the one defined above it is possible to create $r' = u_3\#u_4\$u_1\#u_2$ so that:

$$\begin{aligned}(y, x) \vdash_{r'} (w, z) \quad \text{iff} \quad & x = x_1u_1u_2x_2, y = y_1u_3u_4y_2, \\ & z = x_1u_1u_4y_2, w = y_1u_3u_2x_2, \\ & \text{for some } x_1, x_2, y_1, y_2 \in V^*\end{aligned} \tag{2}$$

where $x, y, z, w, u_1, u_2, u_3, u_4 \in V^*$.

Based on 2-splicing the notion of an *H scheme* can be defined as a pair $\sigma = (V, R)$ where $V$ is an alphabet and $R \subseteq V^*\#V^*\$V^*\#V^*$ is a set of splicing rules. For an H scheme and a language $L \subseteq V^*$ we define

$$\begin{aligned} \sigma(L) &= \{z \in V^* \mid (x, y) \vdash_r (z, w) \text{ or } (x, y) \vdash_r (w, z), \\ &\qquad \text{for some } x, y \in L, r \in R, w \in V^*\}, \\ \sigma^0(L) &= L, \\ \sigma^{i+1}(L) &= \sigma^i(L) \cup \sigma(\sigma^i(L)),\ i \geq 0, \\ \sigma^*(L) &= \bigcup_{i \geq 0} \sigma^i(L). \end{aligned}$$

The *diameter* of $\sigma$ (the concept of diameter was introduced in [3] where it was called width) is indicated by $dia(\sigma) = (n_1, n_2, n_3, n_4)$, where

$$n_i = \max\{|u_i| \mid u_1\#u_2\$u_3\#u_4 \in R\},\ 1 \leq i \leq 4. \tag{3}$$

If we consider two families of languages $FL_1$ and $FL_2$, we define:

$$H(FL_1, FL_2) = \{\sigma^*(L) \mid L \in FL_1 \wedge \sigma = (V, R), R \in FL_2\}.$$

We denote by $FIN, REG$ the families of finite and of regular languages respectively. We have (see details in [9])

$$FIN \subset H(FIN, FIN) \subset REG.$$

An *extended H system* is a construct $\gamma = (V, T, A, R)$, where $V$ and $T$ are alphabets so that $T \subseteq V$ ($T$ is called *terminal* alphabet), $A$ is a language on $V$ ($A$ is the set of *axioms*), and $R$ is a set of splicing *rules* over $V$. The language generated by $\gamma$ is $L(\gamma) = \sigma^*(A) \cap T^*$. The diameter of an extended H system $\gamma$ (indicated by $dia(\gamma) = (n_1, n_2, n_3, n_4)$) is defined in a way similar to (3).

It is known by [1] and [12] that extended H systems with finite sets of axioms and splicing rules characterize $REG$.

A *splicing P system* of *degree* $m, m \geq 1$, is a construct

$$\Pi = (V, T, \mu, L_1, \cdots, L_m, R_1, \cdots, R_m),$$

where $V$ is an alphabet; $T \subseteq V$ is the terminal alphabet; $\mu$ is a tree-like membrane structure consisting of $m$ membranes labeled in a one-to-one manner with $1, \cdots, m$; $L_i \subseteq V^*, 1 \leq i \leq m$ are languages associated with the regions $1, \cdots, m$ of $\mu$; $R_i, 1 \leq i \leq m$, are finite sets of evolution rules associated with the regions $1, \cdots, m$ of $\mu$, of the following form: $(r, tar_1, tar_2)$, where $r = u_1\#u_2\$u_3\#u_4$ is a 2-splicing rule over $V$, $\#, \$ \notin V$ and $tar_1, tar_2 \in \{here, out, in\}$ are called *target indication*.

A *configuration* of $\Pi$ is an $m$-tuple $(M_1, \cdots, M_m)$ of languages over $V$. For two configurations $(M_1, \cdots, M_m), (M'_1, \cdots, M'_m)$ of $\Pi$ we write $(M_1, \cdots, M_m) \Rightarrow (M'_1, \cdots, M'_m)$ if it is possible to pass from $(M_1, \cdots, M_m)$ to $(M'_1, \cdots, M'_m)$ applying in parallel the splicing rules of each membrane of $\mu$ to all possible strings of the corresponding membrane. So for $0 \leq i \leq m$ if $x = x_{i1}u_{i1}u_{i2}x_{i2}, y = y_{i1}u_{i3}u_{i4}y_{i2} \in M_i$ and $(r = u_{i1}\#u_{i2}\$u_{i3}\#u_{i4}, tar_{i1}, tar_{i2}) \in R_i, x_{i1}, x_{i2}, y_{i1}, y_{i2}, u_{i1}, u_{i2}, u_{i3}, u_{i4} \in V^*$, we have $(x, y) \vdash_r (z, w)$, $z, w \in V^*$. The strings $z$ and $w$ will go to the regions indicated by $tar_{i1}$ and $tar_{i2}$ respectively. For $j = 1, 2$, if $tar_{ij} = here$ then the string remains in membrane $i$; if $tar_{ij} = out$ the string is moved to the region immediately above membrane $i$ (if $i$ if the skin membrane the string leaves the system); if $tar_{ij} = in$ the string is moved to any region immediately below in membrane $i$. Note that as strings are supposed to appear in arbitrary many copies, after the application of rule $r$ in a membrane $i$ the strings $x$ and $y$ are still available in the same region. If a string is sent out of a membrane then no copy of it remains here.

A *computation* is a sequence of transitions between configurations of a system $\Pi$ starting from the initial configuration $(L_1, \cdots, L_m)$. The result of a computation is given by all strings in $T^*$ the skin membrane sends out. All strings of this type define the language generated by $\Pi$ and it is indicated by $L(\Pi)$.

Note that if a string is sent out of the system but it is not entirely made of symbols in $T$ it is ignored, on the other hand a string in the system composed only by symbols in $T$ does not contribute to the generated language.

The *depth* of a P system is defined by the height of the tree (i.e. the number of nodes of the longest path from the root to a leaf) describing its membrane structure.

The *diameter* of a splicing P system $\Pi = (V, T, \mu, L_1, \cdots, L_m, R_1, \cdots, R_m)$, indicated by $dia(\Pi) = (n_1, n_2, n_3, n_4)$, is defined by

$$n_i = \max\{|u_i| \mid (u_1\#u_2\$u_3\#u_4; tar_1, tar_2) \in R_1 \cup \cdots \cup R_m, \\ tar_1, tar_2 \in \{here, in, out\}\},\ 1 \leq i \leq 4. \quad (4)$$

We denote by $SPL(i/o, m, p, (n_1, n_2, n_3, n_4))$ the family of languages $L(\Pi)$ generated by splicing P systems as above of degree at most $m, m \geq 1$, depth $p, p \geq 1$ and diameter $(n_1, n_2, n_3, n_4)$.

It is possible to generalize the description of a P system passing from a tree structure to a graph (different from a tree) structure. An *asymmetric planar graph* is so made that for each two nodes $i, j$ there is at most one of $(i, j), (j, i)$ edge. Such a graph is a representation of a planar map such that each border segment can be crossed in one direction only.

A *splicing P system on asymmetric graph* of degree $m, m \geq 1$, is a construct

$$\Pi = (V, T, g, L_1, \cdots, L_m, R_1, \cdots, R_m),$$

where $V, T, L_1, \cdots, L_m, R_1, \cdots, R_m$ are similar to the ones defined for a splicing P system of degree $m$. The only difference is that $tar_{ij} \in \{here, out, go\}, 1 \leq i \leq m, j = 1, 2$, where *here* and *out* have the same effect as described for splicing P systems, and *go* indicates that the string must go to another membrane non-deterministically chosen among the ones to which the string can according to what in indicated in $g$. The set $g$ defines couples indicating the edges of the graph having $L_1, \cdots, L_m$ as nodes. So $g$ defines the permitted communication between the membranes in $\Pi$.

The *diameter* of a splicing P system on asymmetric graph $\Pi$ (indicated by $dia(\Pi)$) is defined in a way similar to (4).

We denote by $SP'L(go, m, (n_1, n_2, n_3, n_4))$ the family of languages $L(\Pi)$ generated by splicing P systems on asymmetric graph as above of degree at most $m, m \geq 1$, and diameter $(n_1, n_2, n_3, n_4)$.

In the next two sections we demonstrate theorems regarding the generative power of splicing P systems and splicing P systems on asymmetric graphs. These theorems represent an improvement of results present in [11] and [4].

# 3 Splicing P systems

In [11] the authors demonstrate that $SPL(i/o, 3, 3, (1, 2, 2, 1)) = RE$ (Theorem 1) and that $SPL(i/o, 5, 2, (1, 2, 2, 1)) = RE$ (Theorem 3). In [4] the authors show that $SPL(i/o, 2, 2, (2, 2, 2, 2)) = RE$ (Theorem 1). Hereby, using a variant of the "rotate-and-simulate" technique introduced in [8], we demonstrate that it is possible to have a splicing P system generating $RE$ keeping the degree of the system equal to 2 (so as a consequence also the depth is 2) and the diameter equal to (1, 2, 2, 1).

**Theorem 1** $SPL(i/o, 2, 2, (1, 2, 2, 1)) = RE$

*Proof.* Let $G = (N, T, S, R)$ be a type-0 Chomsky grammar in Kuroda normal form (this means that the productions in $R$ can be of the forms $A \to a, A \to CD, AC \to DE$ or $A \to \lambda$ where $A, C, D, E \in N$ and $a \in T$) and $B$ be a symbol not in $N \cup T$. Let us assume that symbols in $N \cup T \cup \{B\}$ can be numbered in a one-to-one manner so that $N \cup T \cup \{B\} = \{\alpha_1, \cdots, \alpha_n\}$ and that $R$ contains $m$ productions: $u_i \to v_i, 1 \leq i \leq m$. Moreover $R$ can be divided in two sets: $R_1 = \{u_i \to v_i \mid u_i \to v_i \in R \wedge |u_i| = 1\}$ and $R_2 = \{u_i \to v_i \mid u_i \to v_i \in R \wedge |u_i| = 2\}$ so that $R_1 \cup R_2 = R$ and $R_1 \cap R_2 = \emptyset$. Consider also $R' = \{u \to u \mid u \in \{\alpha_1, \cdots, \alpha_n\}\}$ and that $\{o, X, X_1, X_2, Y, Y_1, Y_2, Z_X, Z_{X_1^+}, Z_{X_2^+}, Z_Y, Z_{Y_1^+}, Z_{Y_2^+}, Z_\lambda, Z'_\lambda\} \cup \{Z_{X_i}, Z_{Y_i} \mid 1 \leq i \leq n + m\} \cup \{Y'_i, Z_{Y'_i} \mid u_i \to v_i \in R_2\}$ are symbols not in $N \cup T$.

Hereby the splicing P system of degree 2, depth 2 and diameter (1, 2, 2, 1) simulating the just defined grammar is described. For a better understanding of the demonstration splicing rules are numbered.

$\Pi = \{V, T, \mu, L_1, L_2, R_1, R_2\}$,
$V = N \cup T \cup \{o, B, X, X_1, X_2, Y, Y_1, Y_2, Z_X, Z_{X_1^+}, Z_{X_2^+}, Z_Y, Z_{Y_1^+}, Z_{Y_2^+}, Z_\lambda, Z'_\lambda\} \cup$
$\{Z_{X_i}, Z_{Y_i} \mid 1 \leq i \leq n + m\} \cup \{Y'_i, Z_{Y'_i} \mid u_i \to v_i \in R_2\}$,
$\mu = [_1[_2]_2]_1$,
$L_1 = \{XBSY, X_2 Z_{X_2^+}, Z_{Y_1^+} Y_1, X Z_X, Z_\lambda, Z'_\lambda\} \cup \{Z_{Y_i} o^i Y_1 \mid 1 \leq i \leq n + m\} \cup$
$\{Z'_{Y_i} Y'_i \mid u_i \to v_i \in R_2\}$,
$L_2 = \{Z_{Y_2^+} Y_2, X_1 Z_{X_1^+}, Z_Y Y\} \cup \{X_1 o^i v_i Z_{X_i} \mid 1 \leq i \leq n + m\}$,

$R_1$={1)$(\#u_iY\$Z_{Y_i}\#; in, out) \mid 1 \leq i \leq n+m\} \cup$
$\{2)(\#CY\$Z_{Y_i'}\#; here, out), 3)(\#AY_i'\$Z_{Y_i}\#; in, out) \mid u_i \to v_i \in R_2\} \cup$
$\{4)(\#Z_{X_2^+}\$X_1\#o; here, out), 5)(\#oY_2\$Z_{Y_1^+}\#; in, out), 6)(\#Z_X\$X_1\#\alpha; in, out),$
$7)(\#BY\$Z_\lambda\#; here, out), 8)(\#Z_\lambda'\$X\#; out, out) \mid \alpha \in N \cup T \cup \{B\}\},$
$R_2$={9)$(\#Y_1\$Z_{Y_2^+}\#; here, out), 10)(\#Z_{X_i}\$X\#; out, out), 11)(\#Z_{X_1^+}\$X_2o\#; out, out),$
$12)(\alpha\#Y_2\$Z_Y\#; out, out) \mid 1 \leq i \leq n+m, \alpha \in N \cup T \cup \{B\}\}.$

During the subsequent demonstration note that all second output strings do not have any active role in the system, so $\Pi$ could be based on 1-splicing.

The idea of the proof is based on the "rotate-and-simulate" technique, classic in H systems area. The sentential forms generated by $G$ are simulated in $\Pi$ in a circular permutation $Xw_1Bw_2Y, w_1, w_2 \in \{N \cup T\}^*$, with variants of $X$ and $Y$. They will be present in a membrane of $\Pi$ if and only if $w_2w_1$ is a sentential form of $G$. It is possible to remove the nonterminal symbol $Y$ only with $B$ from strings of the form $XwBY$, $w \in \{N \cup T\}^*$. In this way the correct permutation of the string is ensured.

The simulation of a production in $R$ and the rotation are done in the same way.

Assume that in membrane 1 we have a string of the form $Xwu_iY$ with $w, u_i \in \{N \cup T \cup \{B\}\}^*$ (initially we have $XBSY$).

If a production in $R_1 \cup R'$ is simulated we have $(Xw \mid u_iY, Z_{Y_i} \mid o^iY_1) \vdash_1 (Xwo^iY_1, Z_{Y_i}u_iY)$ the first output string is sent into membrane 2 while the second is sent out of the system.

If a production in $R_2$ is simulated we have $(XwA \mid CY, Z_{Y_i'} \mid Y_i') \vdash_2 (XwAY_i', Z_{Y_i'}CY)$ (the first output string remains in membrane 1 and the second leaves the system) and then $(Xw \mid AY_i', Z_{Y_i} \mid o^iY_1) \vdash_3 (Xwo^iY_1, Z_{Y_i}AY_i'), 1 \leq i \leq n+m$ (the first output string is sent to membrane 2 and the second leaves the system).

In both cases the suffix $u_iY$ is changed with $o^iY_1, 1 \leq i \leq n+m$. The strings leaving the system do not belong to $T^*$ so they do not contribute to the language generated by $\Pi$.

In membrane 2, with a string as $Xwo^iY_1$, it is possible to perform $(Xwo^i \mid Y_1, Z_{Y_2^+} \mid Y_2) \vdash_9 (Xwo^iY_2, Z_{Y_2^+}Y_1)$. The second output string is sent to membrane 1 where no splicing rule can be applied; the string $Xwo^iY_2$, remaining in membrane 2, can be spliced so to have $(X_1o^jv_j \mid Z_{X_j}, X \mid wo^iY_2) \vdash_{10} (X_1o^jv_jwo^iY_2, XZ_{X_j}), 1 \leq j \leq n+m$. Both output strings are sent to membrane 1 but only the first one can be involved in splicing operations. A string as $Xwo^iY_1$ can also be spliced in membrane 2 by rule 10 so to have: $(X_1o^jv_j \mid Z_{X_j}, X \mid wo^iY_1) \vdash_{10} (X_1o^jv_jwo^iY_1, XZ_{X_j}), 1 \leq j \leq n+m$. Both output strings are sent to membrane 1. The second one cannot be involved in any splicing, with the first it is possible to have $(X_2 \mid Z_{X_2}, X_1 \mid o^jv_jwo^iY_1) \vdash_4 (X_2o^jv_jwo^iY_1, X_1Z_{X_2})$ but both strings, remaining in membrane 1, are no longer spliced.

A string of the form $X_1o^jv_jwo^iY_2$ can be spliced in membrane 1 so to substitute $X_1$ with $X_2$ and $oY_2$ with $Y_1$. This happens by $(X_2 \mid Z_{X_2^+}, X_1 \mid o^jv_jwo^iY_2) \vdash_4 (X_2o^jv_jwo^iY_2, X_1Z_{X_2^+})$ (the first output string remains in membrane 1 while the second is sent out of the system) and $(X_2o^jv_jwo^{i-1} \mid oY_2, Z_{Y_1} \mid Y_1) \vdash_5 (X_2o^jv_jwo^{i-1}Y_1, Z_{Y_1}oY_2)$ (the first output string is sent in membrane 2 while the second leaves the system). In membrane 1 it is also possible to have $(X_1o^jv_jwo^{i-1} \mid oY_2, Z_{Y_1} \mid Y_1) \vdash_5 (X_1o^jv_jwo^{i-1}Y_1, Z_{Y_1}oY_2)$. The second output string is sent out of the system while the first to membrane 2. Here this last string can be spliced so to have $(X_1o^jv_jwo^{i-1} \mid Y_1, Z_{Y_2^+} \mid Y_2) \vdash_9 (X_1o^jv_jwo^{i-1}Y_2, Z_{Y_2^+}Y_1)$. The first output string remains in membrane 2, the second is sent to membrane 1 and both cannot be involved in any splicing operation. The strings sent out of the system do not belong to $T^*$ so they do not contribute to the language generated by $\Pi$.

In membrane 2 a string as $X_2o^jv_jwo^{i-1}Y_1$ can be spliced so to substitute $Y_1$ with $Y_2$ and $X_2o$ with $X_1$. This is obtained by $(X_2o^jv_jwo^{i-1} \mid Y_1, Z_{Y_2^+} \mid Y_2) \vdash_9 (X_2o^jv_jwo^{i-1}Y_2, Z_{Y_2^+}Y_1)$ (the first string remains in membrane 2, the second is sent to membrane 1 and cannot be involved in any splicing) and $(X_1 \mid Z_{X_1^+}, X_2o \mid o^{j-1}v_jwo^{i-1}Y_2) \vdash_{11} (X_1o^{j-1}v_jwo^{i-1}Y_2, X_2oZ_{X_1^+})$ (both output strings are sent to membrane 1 but only the first one can be spliced). In membrane 2 it is also possible to have $(X_1 \mid Z_{X_1^+}, X_2o \mid o^{j-1}v_jwo^{i-1}Y_1) \vdash_{11} (X_1o^{j-1}v_jwo^{i-1}Y_1, X_2oZ_{X_1^+})$. Both strings are sent to membrane 1 but only the first one can be spliced with $X_2Z_{X_2}$ by rule 4 so to obtain $X_2o^{j-1}v_jwo^{i-1}Y_1$, remaining in membrane 1 and no more spliced, and $X_1Z_{X_2}$ not in $T^*$ sent out of the system.

The process of decreasing the number of $o$'s on the left and on the right of strings goes on between membranes 1 and 2. At a certain point three kinds of strings can be present: $X_1v_jwY_2, X_1o^kv_jwY_2$ in membrane 1 and $X_2v_jwo^kY_1$ in membrane 2, $1 \leq k \leq n+m-1$.

As described before a string as $X_1o^kv_jwY_2$ can be spliced with $X_2Z_{X_2^+}$ by rule 4 so to obtain $X_2o^kv_jwY_2$, remaining in membrane 1 and no more spliced, and $X_1Z_{X_2^+} \notin T^*$ sent out of the system.

In membrane 2 a string as $X_2v_jwo^kY_1$ can change the suffix $Y_1$ with $Y_2$ by rule 9 and the string $Z_{Y_2^+}Y_2$. The output strings $X_2v_jwo^kY_2$, remaining in membrane 2, and $Z_{Y_2^+}Y_1$, sent in membrane 1, are no longer used.

The string $X_1v_jwY_2$ can be spliced in membrane 1 so that $(X \mid Z_X, X_1 \mid v_jwY_2) \vdash_6 (Xv_jwY_2, X_1Z_X)$. The first output string is sent to membrane 2 while the second (not in $T^*$) out of the system. In membrane 2 it is possible to have $(Xv_jw \mid Y_2, Z_Y \mid Y) \vdash_{12} (Xv_jwY, Z_YY_2)$. Both output strings are sent to membrane 1 but only the first one can be involved in splicing operations.

What it was just described is the process to pass from $Xwu_iY$ to $Xv_jwY$ simulating a production in $R$ or rotating the substring between $X$ and $Y$ with one symbol.

At any moment a string of the form $XwY$ can be spliced in membrane 1 by rules 7 and 8.

If $(\mid Z'_\lambda, X \mid wY) \vdash_8 (wY, XZ'_\lambda)$ is performed, the first output string, sent out of the system, does not contribute to the language generated by $\Pi$ as $Y \notin T$; the second output string remains in membrane 1 and cannot be involved in any splicing.

If $w = xB, x \in \{N \cup T\}^*$ then $(Xx \mid BY, Z_\lambda \mid) \vdash_7 (Xx, Z_\lambda BY)$ can be performed. The first output string, remaining in the same membrane, can be involved in $(\mid$

$Z'_\lambda, X \mid x) \vdash_8 (x, XZ'_\lambda)$. The strings $x, Z_\lambda BY$ and $XZ'_\lambda$ are sent out of the system but only $x$ can contribute to the language generated by $\Pi$.

If $x \in T^*$ the system $\Pi$ has simulated a derivation of $G$.

In the initial configuration of membrane 2 no splicing can be performed.

As just demonstrated all derivations in $G$ can be simulated in $\Pi$ and, conversely, all correct computations in $\Pi$ correspond to correct derivations in $G$. As we only collect strings in $T^*$ leaving the system $\Pi$, we have $L(G) = L(\Pi)$ proving the theorem.

Considering the definitions (2) and (4) it is easy to see that $SPL(i/o, 2, 2, (2, 1, 1, 2)) = RE$. The proof is similar to the one of Theorem 1 where for each rule the target indications are switched.

## 4 P systems on asymmetric graphs

By $SP'L(go, *)$ we denote the union of all families $SP'L(go, m), m \geq 1$. In [11] the authors demonstrate that $SP'L(go, *) = RE$ (Theorem 9). Hereby we improve this result demonstrating that $SP'L(go, 3) = RE$ and, considering that $SP'L(go, 1) = SP'L(go, 2) = REG$ (Theorem 7 in [11]), our result is minimal concerning the number of membranes used.

A simple way to prove that $SP'L(go, 3, (1, 2, 2, 1)) = SP'L(go, 3, (2, 1, 1, 2)) = RE$ is using Theorem 1. If we consider the graph and the planar map represented in Figure 1 we can imagine that membranes 1 and 2 have the same languages and similar set of evolution rules of membranes 1 and 2 (respectively) present in Theorem 1. Membrane 3 is only used to pass strings from membrane 2 to membrane 1 without changing them. Each splicing rule present in Theorem 1 and containing *in* as target indication is present in the P system on asymmetric graph with *go* instead of *in*, the other target indications are not changed.

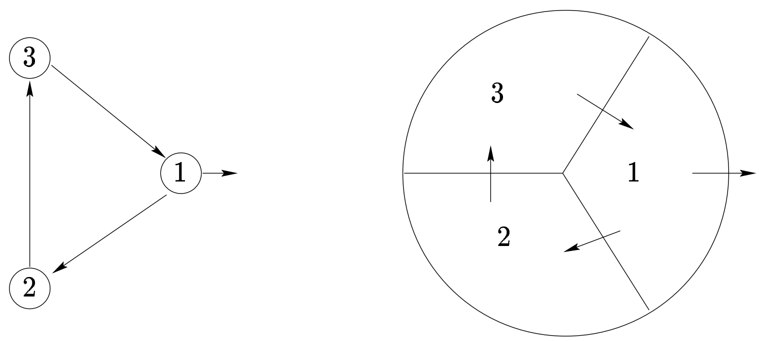


Figure 1: Graph system and planar map in the proof of Theorem 2

The language associated with membrane 3 is $\{Z\}$ and the set of evolution rules is $\{13)(\alpha\#\$Z\#; go, here) \mid \alpha \in \{Y, Y_1, Y_2, Z_{X_i} \mid 1 \leq i \leq n+m\}\}$. The passage of strings from membrane 2 to membrane 1 is made through membrane 3: the first output string is sent to membrane 1, the second, $Z$, remaining in membrane 3, belongs to its language. No splicing is possible in the initial configuration of membrane 3.

Keeping the number of membranes equal to 3 it is possible to reduce the diameter of a P system on asymmetric graph generating $RE$.

**Theorem 2** $SP'L(go, 3, (0, 2, 1, 0)) = SP'L(go, 3, (1, 0, 0, 2)) = RE.$

*Proof.* We only prove that $SP'L(go, 3, (0, 2, 1, 0)) = RE$, the other equality can be obtained using this proof and definitions (2) and (4).

Let $G = (N, T, S, R)$ be a type-0 Chomsky grammar in Kuroda normal form (this means that the productions in $R$ can be of the form $A \to a, A \to CD, AC \to DE$ or $A \to \lambda$ where $A, C, D, E \in N$ and $a \in T$) and $B$ be a symbol not in $N \cup T$. Let us assume that symbols in $N \cup T \cup \{B\}$ can be numbered in a one-to-one manner so that $N \cup T \cup \{B\} = \{\alpha_1, \cdots, \alpha_n\}$ and that $R$ contains $m$ productions: $u_i \to v_i, 1 \leq i \leq m$. Moreover $R$ can be divided in two sets: $R_1 = \{u_i \to v_i \mid u_i \to v_i \in R \ \wedge |u_i| = 1\}$ and $R_2 = \{u_i \to v_i \mid u_i \to v_i \in R \ \wedge |u_i| = 2\}$ so that $R_1 \cup R_2 = R$ and $R_1 \cap R_2 = \emptyset$. Consider also $R' = \{u \to u \mid u \in \{\alpha_1, \cdots, \alpha_n\}\}$ and that $\{X, X', Y, Y', Z_X, Z_{X'}, Z_Y, Z_{Y'}, Z_\lambda, Z'_\lambda\} \cup \{X_i, Y_i, Z_i, Z_{X_i}, Z_{Y_i} \mid 1 \leq i \leq n + m\} \cup \{Y'_i, Z_{Y'_i} \mid u_i \to v_i \in R_2\}$ are symbols not in $N \cup T$.

Hereby the P system on asymmetric graph of degree 3 and diameter (0, 2, 1, 0) simulating the just defined grammar is described. For a better understanding of the demonstration splicing rules are numbered.

$$
\begin{aligned}
\Pi=&\{V, T, g, L_1, L_2, L_3, R_1, R_2, R_3\},\\
V=&N \cup T \cup \{B, X, X', Y, Y', Z_X, Z_{X'}, Z_Y, Z_{Y'}, Z_\lambda, Z'_\lambda\} \cup \\
&\{X_i, Y_i, Z_i, Z_{X_i}, Z_{Y_i} \mid 1 \leq i \leq n + m\} \cup \{Y'_i, Z_{Y'_i} \mid u_i \to v_i \in R_2\},\\
g=&\{(1, 2), (2, 3), (3, 1)\},\\
L_1=&\{XBSY, X'Z_{X'}, Z_\lambda, Z'_\lambda\} \cup \{Z_{Y_i}Y_i \mid 1 \leq i \leq n + m\} \cup \\
&\{X_iZ_{X_i} \mid 1 \leq i \leq n + m - 1\} \cup \{Z_{Y'_i}Y'_i \mid u_i \to v_i \in R_2\},\\
L_2=&\{Z_{Y'}Y'\} \cup \{X_iv_iZ_i \mid 1 \leq i \leq n + m\} \cup \{Z_{Y_i}Y_i \mid 1 \leq i \leq n + m - 1\},\\
L_3=&\{XZ_X, Z_YY\} \cup \{X_iZ_{X_i} \mid 2 \leq i \leq n + m\},\\
R_1=&\{1)(\#u_iY\$Z_{Y_i}\#; go, out) \mid 1 \leq i \leq n + m\} \cup \\
&\{2)(\#CY\$Z_{Y'_i}\#; here, out), 3)(\#AY'_i\$Z_{Y_i}\#; go, out) \mid u_i \to v_i \in R_2\} \cup \\
&\{4)(\#Z_{X_{i-1}}\$X_i\#; go, out) \mid 2 \leq i \leq n + m\} \cup \\
&\{5)(\#Z_{X'}\$X_1\#; go, out), 6)(\#BY\$Z_\lambda\#; here, out), 7)(\#Z'_\lambda\$X\#; out, out)\},\\
R_2=&\{8)(\#Z_i\$X\#; go, go) \mid 1 \leq i \leq n + m\} \cup \\
&\{9)(\#Y_i\$Z_{Y_{i-1}}\#; go, go) \mid 2 \leq i \leq n + m\} \cup \{10)(\#Y_1\$Z_{Y'}\#; go, go)\},\\
R_3=&\{11)(\#Z_{X_i}\$X_i\#; go, here) \mid 2 \leq i \leq n + m\} \cup \\
&\{12)(\#Z_X\$X'\#; go, go), 13)(\#Y'\$Z_Y\#, here, go\}
\end{aligned}
$$

The idea of the proof is again based on the "rotate-and-simulate" technique. The sentential forms generated by $G$ are simulated in $\Pi$ in a circular permutation $Xw_1Bw_2Y, w_1, w_2 \in \{N \cup T\}^*$, with variants of $X$ and $Y$. They will be present in a membrane of $\Pi$ if and only if $w_2w_1$ is a sentential form of $G$. It is possible to remove the nonterminal symbol $Y$ only with $B$ from strings of the form $XwBY$, $w \in \{N \cup T\}^*$. In this way the correct permutation of the string is ensured.

The simulation of a production in $R$ and the rotation are done in the same way.

Assume that in membrane 1 we have a string of the form $Xwu_iY$ with $w, u_i \in \{N \cup T \cup \{B\}\}^*$ (initially we have $XBSY$).

If a production in $R_1 \cup R'$ is simulated we have $(Xw \mid u_iY, Z_{Y_i} \mid Y_i) \vdash_1 (XwY_i, Z_{Y_i}u_iY)$ the first output string is sent into membrane 2 while the second is sent out of the system.

If a production in $R_2$ is simulated we have $(XwA \mid CY, Z_{Y_i'} \mid Y_i') \vdash_2 (XwAY_i', Z_{Y_i'}CY)$ (the first output string remains in membrane 1 and the second leaves the system) and then $(Xw \mid AY_i', Z_{Y_i} \mid Y_i) \vdash_3 (XwY_i, Z_{Y_i}AY_i')$ (the first output string is sent to membrane 2 and the second leaves the system).

In both cases the suffix $u_iY$ is changed with $Y_i, 1 \leq i \leq n+m$. The strings leaving the system do not belong to $T^*$ so they do not contribute to the language generated by $\Pi$.

In membrane 2, with a string as $XwY_i$, it is possible to perform $(X_jv_j \mid Z_j, X \mid wY_i) \vdash_8 (X_jv_jwY_i, XZ_j)$ (for some $1 \leq j \leq n+m$), and both strings are sent to membrane 3, where only the first can be involved in splicing operations.

A string as $X_jv_jwY_i$ is spliced so to decrease the value of the subscripts of $X$ and $Y$ until special situations are present. The subscript of $X$ is decreased in membrane 1, the one of $Y$ in membrane 2; membrane 3 is simply used to pass strings during this process.

So when a string of the form $X_jv_jwY_i, 2 \leq j \leq n+m$ is present in membrane 3 it is moved to membrane 1 by $(X_j \mid Z_{X_j}, X_j \mid v_jwY_i) \vdash_{11} (X_jv_jwY_i, X_jZ_{X_j})$. The string $X_jZ_{X_j}$, remaining in membrane 3, belongs to its language.

In membrane 1 it is possible to have $(X_{j-1} \mid Z_{j-1}, X_j \mid v_jwY_i) \vdash_4 (X_{j-1}v_jwY_i, X_jZ_{j-1})$. The first output string is sent to membrane 2, the second leaves the system (but do not contributes to the language generated by $\Pi$ as it is not in $T^*$).

A string as $X_{j-1}v_jwY_i$ can be spliced in membrane 2 so to have $(X_{j-i}v_jw \mid Y_i, Z_{i-1} \mid Y_{i-1}) \vdash_9 (X_{j-i}v_jwY_{i-1}, Z_{i-1}Y_i)$, both output strings are sent to membrane 3 but the second one cannot be involved in any splicing.

Decreasing the subscripts of $X$ and $Y$ it is possible to have: $X_1v_jwY_k$ in membrane 1 and $X_kv_jwY_1$ in membrane 2, where $2 \leq k \leq n+m$. Considering that $X_1$ and $Y_1$ can be substituted by $X'$ and $Y'$ respectively by rules 5 and 10, it is also possible to have $X'v_jwY'$ in membrane 3.

In the first case $(X' \mid Z_{X'}, X_1 \mid v_jwY_k) \vdash_5 (X'v_jwY_k, X_1Z_{X'})$ is performed. The string $X_1Z_{X'}$ is sent out of the system and do not contributes to the language generated by $\Pi$ as it is not in $T^*$. The first output string is sent to membrane 2 where the subscript of $Y$ is decreased so to have $X'v_jwY_{k-1}$ which is sent to membrane 3. Here $X'$ is substituted with $X$ by $(X \mid Z_X, X' \mid v_jwY_{k-1}) \vdash_{12} (Xv_jwY_{k-1}, X'Z_X)$. Both strings are sent to membrane 1 and no splicing can be performed on them.

In the second case the $Y_1$ in $X_kv_jwY_1$ is substituted with $Y'$ in membrane 2 by $(X_kv_jw \mid Y_1, Z_{Y'} \mid Y') \vdash_{10} (X_kv_jwY', Z_{Y'}Y_1)$ and both output strings are sent to membrane 3. Here only the first one can be involved in a splicing operation changing $Y'$ in $Y$: $(X_kv_jw \mid Y', Z_Y \mid Y) \vdash_{13} (X_kv_jwY, Z_YY')$. The first output string remains in membrane 3, the second is sent to membrane 1. In both cases no splicing can be performed on them.

In the third case two directions of splicing are possible. If $(X \mid Z_X, X' \mid v_j w Y') \vdash_{12} (X v_j w Y', X' Z_X)$ is performed the two output strings are sent to membrane 1 where no splicing rule can be applied on them. If $(X' v_j w \mid Y', Z_Y \mid Y) \vdash_{13} (X' v_j w Y, Z_Y Y')$ is performed the second output string is sent to membrane 1 where no splicing can be performed on it. The string $X' v_j w Y$ remains in membrane 3 where $X'$ can be changed with $X$ by rule 12 so to obtain $X v_j w Y$ and $X' Z_X$ both sent to membrane 1. Here the string $X' Z_X$ cannot be involved in any splicing.

What just described is the process to pass from $X w u_i Y$ to $X v_j w Y$ simulating a production in $R$ or rotating the substring between $X$ and $Y$ of one symbol.

At any moment a string of the form $XwY$ can be spliced in membrane 1 by rules 6 and 7.

If $(\mid Z'_\lambda, X \mid wY) \vdash_7 (wY, XZ'_\lambda)$ is performed the first output string, sent out of the system, does not contribute to the language generated by $\Pi$ as $Y \notin T$; the second output string remains in membrane 1 and cannot be involved in any splicing.

If $w = xB, x \in \{N \cup T\}^*$ then $(Xx \mid BY, Z_\lambda \mid) \vdash_6 (Xx, Z_\lambda BY)$ can be performed. The first output string, remaining in the same membrane, can be involved in $(\mid Z'_\lambda, X \mid x) \vdash_7 (x, XZ'_\lambda)$. The strings $x, Z_\lambda BY$ and $XZ'_\lambda$ are sent out of the system but only $x$ can contribute to the language generated by $\Pi$.

If $x \in T^*$, the system $\Pi$ has simulated a derivation of $G$.

If we consider the three membranes in their initial configurations we can see that the splicing operations that can be performed do not produce any terminal string.

In membrane 1 it is possible to have $(X_{i-1} \mid Z_{X_{i-1}}, X_i \mid Z_{X_i}) \vdash_4 (X_{i-1} Z_{X_i}, X_i Z_{X_{i-1}})$ and $(X' \mid Z_{X'}, X_1 \mid Z_{X_1}) \vdash_5 (X' Z_{X_1}, X_1 Z_{X'})$. In both cases the first output strings are sent to membrane 2 where no splicing can be performed; the second leaves the system but do not contribute to the language generated by $\Pi$ as not terminal.

In membrane 2 the splicing operation $(Z_{Y_i} \mid Y_i, Z_{Y_{i-1}} \mid Y_{i-1}) \vdash_9 (Z_{Y_i} Y_{i-1}, Z_{Y_{i-1}} Y_i)$ generates two strings sent to membrane 3 and no longer used.

In membrane 3 it is possible to have $(X_i \mid Z_{X_i}, X_i \mid Z_{X_i}) \vdash_{11} (X_i Z_{X_i}, X_i Z_{X_i})$. The first output string is sent to membrane 1 while the second, remaining in membrane 3, belongs to its alphabet. In membrane 1 the use of the rule 4 brings to $(X_{i-1} \mid Z_{i-1}, X_i \mid Z_{X_i}) \vdash_4 (X_{i-1} Z_{X_i}, X_i Z_{i-1})$. The first output string is sent to membrane 2 and no longer used; the second leaves the system but do not contribute to the language generated by $\Pi$ as not terminal.

As just demonstrated all derivations in $G$ can be simulated in $\Pi$ and, conversely, all correct computations in $\Pi$ correspond to correct derivations in $G$. As we only collect strings in $T^*$ leaving the system $\Pi$, we have $L(G) = L(\Pi)$ proving the theorem.

## 5 Final remarks

We have considered P systems based on splicing having a tree or a graph as structure. In both cases improvements of theorems demonstrating their generative capability were found. In particular our result concerning splicing P systems on asymmetric graphs is minimal concerning the number of membranes used.

It remains open the question if the diameter used to generate RE families of languages is minimal. Moreover the characterization of families of languages between REG and RE is open to investigations.


## Acknowledgments

I thank the Universitá degli Studi di Milano for its financial support to my PhD and the Universiteit Leiden, personified by Prof. G. Rozenberg, accepting me as PhD student in his friendly group of research.


## References


[1] K. Culik II, T. Harju, Splicing semigroups of dominoes and DNA, *Discrete Appl. Math.*, 31 (1991), 261-277.

[2] T. Head, Formal language theory and DNA; an analysis of the generative capacity of specific recombinant behaviors, *Bull. Math. Biology*, 49 (1987), 737 - 759.

[3] A. Păun, Controlled H systems of small radius, *Fundamenta Informaticae*, 31, 2 (1997), 185 - 193.

[4] A. Păun, M. Păun, On the membrane computing based on splicing, submitted, 2000

[5] Gh. Păun, Computing with membranes. *Journal of Computer and System Sciences*, 61 (2000), and also Turku Centre for Computer Science-TUCS Report No. 208, 1998 http://www.tucs.fi.

[6] Gh. Păun, Computing with membranes. An introduction, *Bulletin of the EATCS*, 67 (Febr. 1999), 139-152.

[7] Gh. Păun, Computing with membranes - A variant: P systems with polarized membranes, *Inter. J. of Foundations of Computer Science*, 11, 1 (2000), 167-182, and *Auckland Univ. CDMTCS Report* No. 089, 1999, http://www.cs.auckland.ac.nz/CDMTCS.

[8] Gh. Păun, Regular extended H systems are computationally universal, *J. Automata, Languages, Combinatorics*, 1, 1 (1996), 27 - 36.

[9] Gh. Păun, G. Rozenberg, A. Salomaa, *DNA Computing*. New Computing Paradigms, Springer-Verlag, Berlin, 1998.

[10] Gh. Păun, Y. Sakakibara, T. Yokomori. P systems on graphs of restricted forms, submitted, 1999.

[11] Gh. Păun, T. Yokomori, Membrane computing based on splicing. In E. Winfree and D. Gifford,
editors, *DNA Based Computers V*. MIT, June 1999,
http://bramble.princeton.edu/DNA5/Tarfiles/paun.tgz. Article accepted to the DIMACS $5^{th}$ International Meeting on DNA Based Computers.

[12] D. Pixton, Regularity of splicing languages, *Discrete Appl. Math.*, 69 (1996), 101-124