# Length P Systems with a Lone Traveler

Artiom Alhazov[1], Rudolf Freund[2], Sergiu Ivanov[3]

[1] Institute of Mathematics and Computer Science
Academy of Sciences of Moldova
Academiei 5, MD-2028, Chişinău, Moldova
artiom@math.md

[2] Faculty of Informatics, Vienna University of Technology
Favoritenstr. 9, 1040 Vienna, Austria
rudi@emcc.at

[3] LACL, Université Paris Est, Creteil, Paris, France
sivanov@colimite.fr

**Summary.** In this paper we consider P systems with linear membrane structures (only one membrane is elementary) with at most one object. We raise and attack the question about the computational power of such systems, depending on the number of membrane labels, kinds of rules used, and some other possible restrictions.

## 1 Introduction

P systems with symbol objects are formal computational models of parallel distributed multiset processing. In the scope of the present research, we only deal with one object, so the model is reduced to sequential distributed tree rewriting controlled by one traveller object with a finite memory. Moreover, we assume the membrane structure to be linear (the tree is a path), with one or two possible membrane labels. Hence, we are interested in controlled rewriting of strings over one or two symbols.

Unbounded linear membrane structures have received the attention of researchers in the past, see, e.g., [4] and [3]. In the latter paper, the authors spoke about the generation *languages* by representing strings $a_1 \cdots a_n$ as labels membranes arranged in a linear structure as

$$[_{a_1} [_{a_2} \cdots [_{a_n} ]_{a_n} \cdots ]_{a_2} ]_{a_1}.$$

A different example of research where unbounded membrane structures played a crucial role for obtaining an important result (the computational completeness of P systems with active membranes without polarizations) is given in [1], improved in terms of presentation and object/symbol/membrane label complexity in [2].

This research direction, focusing on the membrane structure (rather than the multiset of objects in a designated region) as the result of the computation of a

P system, has been recalled during the $12^{th}$ Brainstorming Week on Membrane Computing in Sevilla. The technique proposed there how to generate (the description of) recursively enumerable sets of vectors of non-negative integers, using membrane structures with only two labels (0 and 1), where the number of membranes labeled by 1 remains bounded by a constant throughout the computation, is explained in Section 3. It was also conjectured that:

> With one label and at most one "traveler" we can only characterize linear sets, even with membrane generation and deletion.

We confirm this conjecture in Section 4. Finally, in Section 5 we discuss variants of the model leading to weak computational completeness.

## 2 P Systems with Membrane Creation and Dissolution

A *P system with membrane creation and dissolution* is a construct defined as follows:

$$\Pi = (O, H, \mu, h_1, \cdots, h_n, w_1, \cdots, w_n, R) \text{ where}$$

- $O$ is the (finite) alphabet of *objects*;
- $H$ is the (finite) alphabet of membrane *labels*;
- $\mu$ is the initial *membrane structure* consisting of $n$ membranes labeled with elements of $H$;
- $h_i \in H$, $1 \leq i \leq n$, is the initial label of the membrane $i$;
- $w_i \in O^*$, $1 \leq i \leq n$, is the string which represents the initial contents of membrane $i$;
- $R$ is the set of rules.

The rules in $R$ are of one of the following types:

($b$) $[_{h_2} a[_{h_1} \ ]_{h_1} ]_{h_2} \rightarrow [_{h'_2} [_{h'_1} b]_{h'_1} ]_{h'_2}$,
$a, b \in O$, $h_1, h_2, h'_1, h'_2 \in H$ – *send-in* rule,
($c$) $[_{h_2} [_{h_1} a]_{h_1} ]_{h_2} \rightarrow [_{h'_2} b[_{h'_1} \ ]_{h'_1} ]_{h'_2}$,
$a, b \in O$, $h_1, h_2, h'_1, h'_2 \in H$ – *send-out* rule,
($d$) $[_{h_2} [_{h_1} a]_{h_1} ]_{h_2} \rightarrow [_{h'_2} b]_{h'_2}$,
$a, b \in O$, $h_1, h_2, h'_2 \in H$ – *membrane dissolution* rule,
($e$) $[_{h_1} a]_{h_1} \rightarrow [_{h'_1} [_{h_2} b]_{h_2} ]_{h'_1}$,
$a, b \in O$, $h_1, h_2, h'_1 \in H$ – *membrane creation* rule.

A rule of type ($b$) consumes the symbol $a$ in a membrane with label $h_1$ and puts a symbol $b$ into an inner membrane, rewriting the labels of the involved membranes. Symmetrically, a rule of type ($c$) consumes an instance of $a$ in a membrane with label $h_1$, which is located within a membrane with label $h_2$, and puts an instance of $b$ into the latter membrane, rewriting the labels.

A rule of type $(e)$ consumes an instance of $a$ in membrane with label $h_1$ and adds to it a new membrane with label $h_1$, with an instance of $b$ inside. The label of the original membrane is rewritten to $h'_1$. Symmetrically, a rule of type $(d)$ consumes an instance of $a$ in a membrane with label $h_1$, which is located within a membrane with label $h_2$, copies all the symbols from the membrane with label $h_1$ to its parent membrane, discards the former membrane, adds a $b$ to the latter membrane and rewrites its label to $h'_2$.

The rules are applied in the maximally parallel way, with the restriction that in one derivation step at most one rule of types $(b)$, $(c)$, $(d)$, and $(e)$ can be applied per each membrane labeled by $h_1$ in the definition of the rules given above.

A *configuration* $C_k$ of the system $\Pi$ consists of the description of the membrane structure $\mu_k$, the labeling of the membranes, and the multisets over $O$ representing the contents of the regions. A configuration is called *halting* if no more rules are applicable any more. A computation of $\Pi$ is a sequence of configurations $(C_k)_{1\leq k\leq m}$, where $C_1$ is the initial configuration, $C_m$ is a halting configuration, and $C_{k+1}$ is obtained from $C_k$ by applying the rules from $R$.

*Note*: In the following, we will restrict ourselves to P systems with a linear membrane structure. In such systems, membrane creation should never be applied in a non-elementary membrane, otherwise a non-linear membrane structure would be obtained, which is unwanted in the model of the current paper. In Section 3, the P system is constructed in such a way that the object triggering membrane creation would only appear in the elementary membrane. However, in Sections 4 and 5, the setup is more restricted (e.g., one label only), which would yield too restrictive P systems. To overcome this, we impose the following restriction: the membrane creation rule is disabled in the non-elementary membranes. We refer to this rule kind as $(e_e)$. In a similar way, we are interested in membrane dissolution rules which are disabled in the non-elementary membranes, and we denote them by $(d_e)$. Clearly, for each of these membrane dissolution and creation operations, the constructions in Section 3 work for either variant (emphasizing $(d_e)$, otherwise the decrement could be simplified). The regularity conjecture originally assumed only elementary dissolution to be used as well as results only to be obtained at halting with the object in the elementary membrane.

In Section 5 we also consider the following type of rule (introduced already in [3]; we have removed the object in the inner membrane on the right side to let the systems considered in the paper have at most one object during the computation):

$(f)$ $[_{h_1} a]_{h_1} \rightarrow [_{h'_1} b[_{h_2}\ ]_{h_2}]_{h'_1}$,
$a, b \in O$, $h_1, h_2, h'_1 \in H$ – *membrane duplication* rule:

the membrane $h_1$, in the presence of an object $a$, is duplicated, that is, the label $h_1$ is changed into $h'_1$, the object $a$ is replaced by $b$ and a new inner membrane labeled by $h_2$ is created; all the contents of membrane $h_1$ (membranes or objects except this copy of object $b$) is now inside membrane $h_2$.

*Note*: In [3] the authors have assumed the outer membrane to be the newly created one; it makes no difference as long as we can change both labels by this rule. However, we prefer to view the *inner* membrane as the new membrane. This lets us keep $h_1' = h_1$.

Moreover, also the following simplifications/restrictions are made to the rule types in Section 5: membranes $h_2$, $h_2'$ are not mentioned in the notation of rules $(b)$, $(c)$ and $(d)$, which means that the rules mentioned above act independently of the external membranes and do not modify them:

$(b_r)$ $a[_{h_1}\ ]_{h_1} \to [_{h_1'}\, b\,]_{h_1'}$, $a, b \in O$, $h_1, h_1' \in H$,
$(c_r)$ $[_{h_1}\, a\,]_{h_1} \to b[_{h_1'}\ ]_{h_1'}$, $a, b \in O$, $h_1, h_1' \in H$,
$(d_r)$ $[_{h_1}\, a\,]_{h_1} \to b$, $a, b \in O$, $h_1 \in H$.

## 3 Length P Systems

In what follows we will consider a special class of P systems with membrane creation and dissolution. A *length P system $\Pi$* is a P system with membrane creation and dissolution which has the following properties:

- the membrane structure is *linear* in every configuration, i.e., every membrane has at most one inner membrane;
- the membranes in any halting configuration of $\Pi$ are labeled with two labels only.

Consider a halting configuration $C$ of a length P system $\Pi$ and construct the sequence of membrane labels $(h_i)_{1 \le i \le n}$, in which $h_1$ corresponds to the label of the skin membrane, $h_2$ to the label of the membrane inner to the skin membrane, etc., and $n$ is the number of membranes in $C$. Since $h_i$ is a member of a two-element set, we can interpret this sequence as a vector of numbers coded in unary by runs of one label and separated by instances of the other label. This vector of numbers will be considered as the *output* of the length P system $\Pi$.

Following the same convention, we can define the input of $\Pi$ as a two-label membrane structure coding a certain vector of numbers.

We will now show that length P systems with the input supplied via the membrane structure are computationally complete. To achieve this goal we will pick an arbitrary register machine $\mathcal{M}$ and simulate it with the length P system $\Pi_1$ which only uses two labels $H = \{0, 1\}$, only the skin membrane is not empty in the initial configuration and contains $q_s$, and the sequence of initial membrane labels written as a string $h_1 h_2 \cdots h_n$ has the form $110^{R_1}10^{R_2}1 \cdots 10^{R_m}1$, where $R_i$, $1 \le i \le m$, is the value of the $i$-th register of $\mathcal{M}$.

The evolution of the system starts with the rule

$$[_h\, q_s [_1\ ]_1\, ]_h \to [_h\, [_1\, q_{1,1}\, ]_1\, ]_h, \; 0 \le h \le 1,$$

where the symbol $q_{1,1}$ represents the first instruction in the program of $\mathcal{M}$ and also keeps the information about the fact that it is located in the region of the membrane structure corresponding to the first register of $\mathcal{M}$.

To simulate an increment of the $i$-th register of $\mathcal{M}$, we need to add a membrane to the membrane structure and assure that the sequence of labels changes from $110^{R_1}1\cdots10^{R_i}1\cdots10^{R_m}1$ to $110^{R_1}1\cdots10^{R_i+1}1\cdots10^{R_m}1$. We start with the symbol $q_{l,1}$ in the inner membrane of the skin, where $l$ is the label of the increment operation, and we move it to the innermost membrane, counting the registers we traverse on that way:

$$\begin{aligned}&[_h\, q_{l,j}[_0\ ]_0\,]_h \to [_h\,[_0\, q_{l,j}\,]_0\,]_h,\\ &[_h\, q_{l,j}[_1\ ]_1\,]_h \to [_h\,[_1\, q_{l,j+1}\,]_1\,]_h,\ 1 \le j \le m,\ 0 \le h \le 1.\end{aligned}$$

When we reach the end marker of the last register, we go into the innermost membrane and add a new membrane:

$$[_1\, q_{l,m+1}\,]_1 \to [_1\,[_0\, s_{l,m+1}\,]_0\,]_1.$$

We have now changed the sequence of labels from $110^{R_1}1\cdots10^{R_i}1\cdots10^{R_m}1$ to $110^{R_1}1\cdots10^{R_i}1\cdots10^{R_m}10$.

The series of $s$-symbols will now swap this new membrane label 0 with the labels of inner membranes, in order to obtain the new 0 label in the zone of the membrane structure corresponding to register $R_i$:

$$\begin{aligned}&[_0\,[_0\, s_{l,j}\,]_0\,]_0 \to [_0\, s_{l,j}[_0\ ]_0\,]_0,\\ &[_1\,[_0\, s_{l,j}\,]_0\,]_1 \to [_0\, s_{l,j-1}[_1\ ]_1\,]_0,\ i < j \le m+1.\end{aligned}$$

When we produce the symbol $s_{l,i}$, we have already pushed all labels corresponding to the registers with numbers greater than $i$ towards the innermost membrane and we have added a 0-labeled membrane to the zone corresponding to the $i$-th register. The following rules move the $s$-symbol back into the skin membrane:

$$\begin{aligned}&[_h\,[_0\, s_{l,j}\,]_0\,]_h \to [_h\, s_{l,j}[_0\ ]_0\,]_h,\\ &[_h\,[_1\, s_{l,j}\,]_1\,]_h \to [_h\, s_{l,j-1}[_1\ ]_1\,]_h,\ 1 \le j \le i,\ 0 \le h \le 1.\end{aligned}$$

Finally, $s_{l,0}$ produces the symbol corresponding to the next instruction $l'$ of the register machine:

$$[_h\, s_{l,0}[_1\ ]_1\,]_h \to [_h\,[_1\, q_{l',1}\,]_1\,]_h,\ 0 \le h \le 1.$$

The simulation of a decrement and zero-check of the $i$-th register of $\mathcal{M}$ is symmetric to the simulation of an increment: we start with finding the zone of the membrane structure corresponding to the $i$-th register:

$$\begin{aligned}&[_h\, q_{l,j}[_0\ ]_0\,]_h \to [_h\,[_0\, q_{l,j}\,]_0\,]_h,\\ &[_h\, q_{l,j}[_1\ ]_1\,]_h \to [_h\,[_1\, q_{l,j+1}\,]_1\,]_h,\ 1 \le j < i,\ 0 \le h \le 1.\end{aligned}$$

If the value of the register is zero, the symbol $q_{l,i}$ immediately encounters another membrane with the label 1, so it produces the corresponding signal symbol:

$$[_1\, q_{l,i}[_1\ ]_1\,]_1 \to [_1\,[_1\, z_{l,i+1}\,]_1\,]_1.$$

The signal symbol then bubbles up into the skin membrane:

$$\begin{aligned}&[_h\, [_0\, z_{l,j}\, ]_0\, ]_h \to [_h\, z_{l,j} [_0\ ]_0\, ]_h,\\ &[_h\, [_1\, z_{l,j}\, ]_1\, ]_h \to [_h\, z_{l,j-1} [_1\ ]_1\, ]_h,\ 1 \le j \le i,\ \ 0 \le h \le 1.\end{aligned}$$

Finally, in the outer membranes, $z_{l,0}$ produces the symbol coding the next instruction $l'$, corresponding to unsuccessful decrement:

$$[_h\, z_{l,0} [_1\ ]_1\, ]_h \to [_h\, [_1\, q_{l',1}\, ]_1\, ]_h,\ 0 \le h \le 1.$$

If, however, $q_{l,i}$ detects that the $i$-th register is not empty, it produces a different signal symbol:

$$[_1\, q_{l,i} [_0\ ]_0\, ]_1 \to [_1\, [_0\, d_{l,i}\, ]_0\, ]_1.$$

This symbol moves all membrane labels one step outwards:

$$\begin{aligned}&[_0\, d_{l,j} [_0\ ]_0\, ]_0 \to [_0\, [_0\, d_{l,i}\, ]_0\, ]_0, \quad i \le j \le m,\\ &[_0\, d_{l,j} [_1\ ]_1\, ]_0 \to [_1\, [_0\, d_{l,i+1}\, ]_0\, ]_1,\ i \le j \le m.\end{aligned}$$

When it reaches the end of the zone of the membrane structure corresponding to the $m$-th register, $d_{l,m+1}$ dissolves the innermost membrane:

$$[_1\, [_0\, d_{l,m+1}\, ]_0\, ]_1 \to [_1\, s_{l,m+1}\, ]_1.$$

Now the $s$-symbols go outwards into the skin membrane:

$$\begin{aligned}&[_h\, [_0\, s_{l,j}\, ]_0\, ]_h \to [_h\, s_{l,j} [_0\ ]_0\, ]_h,\\ &[_h\, [_1\, s_{l,j}\, ]_1\, ]_h \to [_h\, s_{l,j-1} [_1\ ]_1\, ]_h,\ 1 \le j \le m+1,\ \ 0 \le h \le 1.\end{aligned}$$

And finally, $s_{l,0}$ generates the symbol coding the next operation $l''$, corresponding to a successful decrement:

$$[_h\, s_{l,0} [_1\ ]_1\, ]_h \to [_h\, [_1\, q_{l'',1}\, ]_1\, ]_h,\ 0 \le h \le 1.$$

*Note*: The construction can be rewritten such that only one membrane participates on the left side of any rule, but then the total number of membranes labeled 1 will no longer remain a constant, but will still stay bounded. In the construction presented above, however, the number of membranes labeled by 1 is constant during the computation, namely, it is $m+2$. In the construction above, membrane creation and membrane dissolution rules do not modify the label of the outer membrane, while the communication rules either keep unchanged the labels of the two membranes they work with, or they swap them. The number of membranes labeled by 1 may be reduced by one, by starting with the skin labeled by 0; this does not affect the proof. Moreover, with a technique mentioned in the end of Section 5 the innermost membrane labeled 1 may be avoided. For the special case $m = 2$, in Section 5 we present a construction when, using membrane duplication, we avoid even the membrane labeled by 1 which separates the representation of the two registers, by keeping track of this position with the object itself.

Before that, in the following section, we proceed with the conjecture that length P systems with one label only generate regular sets of numbers, in case of elementary membrane creation, elementary membrane dissolution and communication rules.

## 4 On the Regularity Conjecture

Clearly, any regular set of numbers can be generated with rules $(e)$ only, simulating each rule $pa \to q$, $p \to q_h$ of a finite automaton (for which, without restricting generality we require that every state except $q_h$ is non-final and has at least one outgoing transition) by rules $[_0\, p\,]_0 \to [_0\, [_0\, q\,]_0\,]_0$ and $[_0\, p\,]_0 \to [_0\, [_0\, q_h\,]_0\,]_0$, respectively. Here, the skin and the elementary membrane are to be seen as additional endmarkers (as otherwise only numbers $\geq 2$ could be generated). The latter can be avoided by additionally using a rule of type $(d)$ $[_0\, [_0\, q_h\,]_0\,]_0 \to [_0\, q'_h\,]_0$.

There are two possible reasons why the power of length P systems with one label and one object is restricted. The first reason (R1) is that the object can never detect that it is in the skin (unless we additionally allow the skin to have a distinguished label). Indeed, if $C \Rightarrow C'$ is one computation step, then it is easy to see (just by looking at all kinds of rules) that

$$[_0\, C\,]_0 \Rightarrow [_0\, C'\,]_0 \tag{1}$$

is also a valid computation step. This immediately generalizes to multiple membrane levels and multiple derivation steps:

$$C \Rightarrow^* C' \longrightarrow \underbrace{[_0 \cdots [_0}_{n} C\,]_0 \cdots\,]_0 \Rightarrow^* \underbrace{[_0 \cdots [_0}_{n} C'\,]_0 \cdots\,]_0. \tag{2}$$

The second reason (R2) is that the total membrane depth can only be increased from the elementary membrane (unless we additionally allow membrane duplication rules). Let us denote by $C \Rightarrow_{\text{dive}} C'$ a "membrane dive", i.e., such a computation fragment that the object is in the elementary membrane in both $C$ and in $C'$, but never in the intermediate configurations. Let us also denote by $\langle n, a\rangle$ the configuration consisting of a linear structure of $n$ membranes, the elementary one containing object $a$:

$$\langle n, a\rangle = \underbrace{[_0 \cdots [_0}_{n} a\,]_0 \cdots\,]_0.$$

Clearly, for any $a, b \in O$, one of the following cases is true:

- There exists some minimal value $n \in \mathbb{N}$ for which $\langle n, a\rangle \Rightarrow_{\text{dive}} \langle n, b\rangle$ is true. Let us denote this value by $n(a, b)$. We recall from (2) that $\langle n, a\rangle \Rightarrow_{\text{dive}} \langle n, b\rangle$ holds for any $n \geq n(a, b)$.
- $\langle n, a\rangle \Rightarrow_{\text{dive}} \langle n, b\rangle$ is not true for any $n \in \mathbb{N}$.

We denote by $\bar{n}$ the maximum of values $n(a, b)$ over the first case. We denote by $N$ the set of all numbers generated by the length P system not exceeding $\bar{n}$. We denote by $A$ the set of all objects $a \in O$ such that $\langle \bar{n}, a\rangle$ is reachable in the length P system.

It is easier to confirm the conjecture for the case when the membrane dissolution is only allowed for the elementary membranes. Then, the power of the length

P system is described by the union of the finite set $N$ with the power (plus number $N$) of a partially blind 1-register machine starting in states from $A$, where the chain rules $a \to b$ correspond to the relation $\langle \bar{n}, a\rangle \to \langle \bar{n}, b\rangle$, and the increment/decrement instructions are associated to the rules creating and dissolving elementary membranes. It is known, see, e.g., [5], that the number sets generated by partially blind 1-register machines is $NMAT = NREG$. Finally, in this case it is immediate that the membrane structure cannot change between the last time the object is in the elementary membrane and the halting.

**Note**: The regularity conjecture remains valid even if non-elementary membranes can be dissolved. Indeed, a similar (non-constructive) argument can be made; we only describe it *informally*. For any two symbols $a, b \in O$, either there exists some minimal number $m \in \mathbb{N}$ such that $\langle m, a\rangle \Rightarrow \langle m+1, b\rangle$, and we denote it by $m(a,b)$, or this derivation is not possible for any $m \in \mathbb{N}$. Then the behaviour of the length P system can be decomposed into a finite part (the membrane structure depth not exceeding the maximum of all defined values $m(a,b)$), and a regular part defined by the binary relation over $O$ which is the domain where $m(a,b)$ is defined. As for the computation between the last time the object is in the elementary membrane and the halting (in case the halting is not in the elementary membrane), this should also preserve regularity, because without being able to test for skin and without returning to the elementary membrane, we just have a finite control walking across the unilabeled membrane structure, and membrane dissolution in this case can only perform some regular erasing transformation.

## 5 Weak Computational Completeness

In this section we show that we can construct length P systems with one label which are weakly computationally complete, assuming the following ingredients:

- Membrane duplication rules are allowed (nullifying reason R2).
- The skin can be distinguished (nullifying reason R1) either by being the only membrane with another label ($s = 1$), or by having its own set of rules $R_s$. We use the first case for the presentation of the result.
- As typical in membrane computing, the dissolution is not limited to elementary membranes.
- Membrane creation rules are disabled in the non-elementary membranes. (Alternatively, if one allows to also have special rules for the elementary membrane, we may forbid membrane creation in the rest of the system).

The result relies on simulating 2-register machines, storing the register values in the multiplicity of membranes, using the object to separate the two numbers.

$$[_1\, \underbrace{[_0 \cdots [_0}_{n_1}\, a\, \underbrace{[_0 \cdots [_0}_{n_2}\, ]_0 \cdots ]_0 ]_0 \cdots ]_0 ]_1 \tag{3}$$

We first present a simpler construction, assuming an additional elementary membrane labeled 1. In this case, membrane creation rules are not even needed.

Indeed, the first register is tested for zero by using the skin rules (duplicate, enter, dissolve). The second register is tested for zero by entering one membrane, checking its label and exiting it.

The increment is done by a duplication rule $(f)$, in case of the first register followed by entering the newly created membrane.

The decrement is performed by a dissolution rule $(d)$, in case of the second register preceded by moving the object into the next membrane.

We proceed to the formal description of the simulation (membrane label $h$ stands for any of 0 or 1).

$(l_1 : A(1), l_2, l_3)$ is performed as:

$[_h\, l_1\, ]_h \to [_h\, l_1'[_0\ ]_0\, ]_h,\ l_1'[_0\ ]_0 \to [_0\, l_2\, ]_0, \qquad l_1'[_0\ ]_0 \to [_0\, l_3\, ]_0.$

$(l_1 : A(2), l_2, l_3)$ is performed as:

$[_h\, l_1\, ]_h \to [_h\, l_2[_0\ ]_0\, ]_h,\ [_h\, l_1\, ]_h \to [_h\, l_3[_0\ ]_0\, ]_h.$

$(l_1 : S(1), l_2, l_3)$ is performed as:

$[_0\, l_1\, ]_0 \to l_2,$

$[_1\, l_1\, ]_1 \to [_1\, l_1'[_0\ ]_0\, ]_1,\ l_1'[_0\ ]_0 \to [_0\, l_1''\, ]_0, \qquad [_0\, l_1''\, ]_0 \to l_3.$

$(l_1 : S(2), l_2, l_3)$ is performed as:

$l_1[_0\ ]_0 \to [_0\, l_1'\, ]_0, \qquad [_0\, l_1'\, ]_0 \to l_2,$

$l_1[_1\ ]_1 \to [_1\, l_1'\, ]_1, \qquad [_1\, l_1'\, ]_1 \to l_3[_1\ ]_1.$

Notice that the first three operations above ($A(1)$, $A(2)$ and $S(1)$) operate identically also in the absence of the elementary membrane labeled 1, and so does the first line of $S(2)$, corresponding to the decrement case. Using membrane creation $(e)$ (immediately followed by dissolution) to test for the membrane elementarity, it is possible to avoid the extra elementary membrane with 1, and stay with the representation (3): we replace the last two rules of the construction above with the following ones:

$$[_h\, l_1\, ]_h \to [_h\, [_0\, l_1''\, ]_0\, ]_h,\ [_0\, l_1''\, ]_0 \to l_3.$$

In this way, using only one label for non-skin membranes, weak computational completeness of length P systems is shown with one object, by taking advantage of rules creating non-elementary membranes under the assumption that elementary membrane creation is disabled in the non-elementary membranes.

## 6 Discussion

We have introduced P systems with a linear membrane structure (i.e., only one membrane is elementary) with at most one object. The result of such systems, called length P systems, is either the total number of membranes at halting, or the vector of numbers of consecutive membranes labeled 0. In Section 3 we presented the simulation of register machines with any fixed number of registers.

The power of length P systems with one object and one membrane label depends on two factors: whether the object can detect being in the skin membrane, and whether non-elementary membrane creation is allowed. The first factor is related to the zero-test of the "first" register, and the second factor is related to the possibility of effectively operating with two numbers instead of one. Since the regularity conjecture assumed that these two ingredients are not allowed, we have *confirmed* the conjecture.

In Section 5 we have shown that removing both of these conditions leads to P systems being weakly computationally complete. Questions arise about the intermediate extensions.

We now formulate the following the following two *conjectures*:

*Conjecture 1.* Length P systems produce only regular languages even when membrane duplication is allowed (i.e., without reason R2). The intuition behind the conjecture is that such P systems relate to 2-register machines, where one register is blind (i.e., it can be incremented and decremented, but cannot be tested for zero, and the machine's computation is discarded without the result if decrement of zero is attempted).

*Conjecture 2.* Length P systems produce only regular languages even if the skin membrane has a distinguished label (i.e., without reason R1). The intuition behind the conjecture is that such P systems should relate to 1-register machines, but for the proof many more cases would have to be investigated.

In an *extended version* of this paper we plan to consider length P systems where the number of membranes labeled by 1 is not bounded, considering binary strings as the results instead of vectors.

**Acknowledgements.** Artiom Alhazov acknowledges project STCU-5384 *Models of high performance computations based on biological and quantum approaches* awarded by the Science and Technology Center in the Ukraine.

## References

1. A. Alhazov: P Systems without Multiplicities of Symbol-Objects. *Information Processing Letters* **100** (3), 2006, 124–129.
2. A. Alhazov, R. Freund, A. Riscos-Nunez: Membrane Division, Restricted Membrane Creation and Object Complexity in P Systems. *International Journal of Computer Mathematics* **83** (7), 2006, 529-548.
3. F. Bernardini, M. Gheorghe: Languages Generated by P Systems with Active Membranes. *New Generation Computing* **22** (4), 2004, 311–329.
4. R. Freund: Special Variants of P Systems Inducing an Infinite Hierarchy with Respect to the Number of Membranes. *Bulletin of the EATCS* **75**, 2001, 209–219.

5. R. Freund, O.H. Ibarra, Gh. Păun, H.C. Yen: Matrix Languages, Register Machines, Vector Addition Systems. Proceedings of the *Third Brainstorming Week on Membrane Computing*, Sevilla, 2005, 155–167.