



# Simple variants of non-cooperative polymorphic P systems

Anna Kuczik<sup>1</sup> · György Vaszil<sup>1</sup>

Received: 17 December 2023 / Accepted: 21 March 2024  
© The Author(s) 2024

## Abstract

We investigate the computational power of non-cooperative polymorphic P systems with no additional ingredients. The variants we study are even more simple in the sense that the sets of possible right-hand sides of the dynamically changing rules are finite. We show that systems with this type of restriction characterize exactly the class of Parikh sets of ETOL languages.

**Keywords** P systems with dynamic rules · Polymorphic P systems · P systems with non-cooperative rules · P systems with limited depth · Parikh sets of ETOL languages

## 1 Introduction

Membrane systems or P systems were introduced in [1] as a computing model based on an abstract view of the processes taking place in living cells. They consist of a nested structure of membranes, and the regions delimited by these membranes contain different multisets of objects. These multisets are transformed step by step during the functioning of the system until the result is produced when the system reaches a halting configuration. In the basic variant of the model, the objects can be rewritten in each computational step by parallel application of multi-set rewriting rules, or they can be transferred (communicated) between neighboring regions. More details on the different variants and their computational properties can be found in the monograph [2] or in the handbook [3].

Besides the static rule sets associated to the regions (as considered in the original variant of the model), several methods were also introduced with the aim of making the rule sets more “dynamic”, that is, with the aim of constructing systems which are able to change not only the objects, but also the rules that are applicable to the objects. One of

the first of these types of models were called generalized P systems, see [4], where the operators acting on the objects could also be changed by the system during the computation. Features like rule creation or the inhibiting/de-inhibiting of rules were also considered in [5] and in [6]. In this work, we focus on another one of these dynamic models called polymorphic P systems, which was introduced in [7].

Polymorphic P systems were motivated by the idea that the program of a computing device could be viewed as data. Therefore, it could also be changed during the course of the computation. In these types of P systems, rules are not statically defined, but are dynamically inferred from the contents of pairs of membranes: The contents of one member of the pair define the multiset representing the left-hand side of the rule, the contents of the other member define the right-hand side. As the membranes can contain further membranes, the contents of the pairs, and this way the left- and right-hand sides of rules may change dynamically during the computation. See the survey [8] for more information.

The initial results presented in [7] show the power of the model. With cooperative rules (rules with left-hand sides having more than one objects) any recursively enumerable set of numbers can be generated, but non-cooperative systems (systems with rules having just one object on the left-hand side) can also generate several interesting languages, mainly based on the fact that exponential, even super-exponential growth of the number of objects inside the system can be produced.

The study of non-cooperative variants of the model was continued further in [9] by considering the case of “no ingredients”, that is, when no special features (not even target

---

Anna Kuczik and György Vaszil have contributed equally to this work.

✉ György Vaszil  
vaszil.gyorgy@inf.unideb.hu

Anna Kuczik  
kuczik.anna@inf.unideb.hu

<sup>1</sup> Faculty of Informatics, University of Debrecen, Kassai út 26, 4028 Debrecen, Hungary

indicators) are added to the system. The equivalence of so-called strong and weak polymorphism was shown, left polymorphism, right polymorphism, and general polymorphism was defined. As its main contribution, [9] presented a hierarchy of computational power based on the depth of the membrane structure, but in general, many questions concerning the computational capabilities of the non-cooperative variant remained open for further investigation.

In the present work, we intend to take some additional steps in this direction. We show that (1) Parikh sets of ETOL languages can be generated using non-cooperative polymorphic P systems (with no other ingredients) of depth three where all non-dynamical rules are “chain rules”, and that (2) ETOL systems can generate string languages corresponding to the multiset languages of non-cooperative polymorphic P system where the set of the possible contents of regions corresponding to right-hand sides of rules is finite. This gives us an exact characterization of the class of Parikh sets of ETOL languages in terms of restricted variants of non-cooperative polymorphic P systems. Polymorphic P systems were already considered in connection with Lindenmayer systems in [10], where so called bounded L systems were simulated by polymorphic P systems with target indicators.

In the following, after reviewing the necessary definitions, we present in Sect. 3 how ETOL systems can be simulated with non-cooperating polymorphic P systems of depth three. Then, in Sect. 4 we consider systems which may have unlimited depth, but a limited complexity in the sense that the behavior of their membranes can be described by finite transition systems. We show that these types of non-cooperating polymorphic systems, regardless of their depth, characterize the class of Parikh sets of languages generated by ETOL systems.

## 2 Preliminaries

In the following, we briefly define the basic notions we will use. See [11] for more on formal language theory, and [2, 3] for details about membrane computing.

An alphabet  $V$  is a finite non-empty set of symbols called letters. A string (or word) over  $V$  is a finite sequence of letters, the set of all strings over  $V$  (the free monoid generated by  $V$ ) is denoted by  $V^*$ , and  $V^+ = V^* \setminus \{\lambda\}$  where  $\lambda$  denotes the empty string. For a string  $w \in V^*$ , we denote by  $|w|_x$  the number of occurrences of the letter  $x \in V$  in  $w$ . If we fix an order  $V = \{a_1, a_2, \dots, a_n\}$  of the letters, then the vector  $(|w|_{a_1}, |w|_{a_2}, \dots, |w|_{a_n})$  is called the Parikh vector of the word  $w \in V^*$ .

Multisets can be thought of as sets with multiplicities associated to their elements. If  $\mathbb{N}$  denotes the set of nonnegative integers, then a *multiset* over a set  $U$  is a

mapping  $M : U \rightarrow \mathbb{N}$  where  $M(a)$ , for all  $a \in U$ , is the multiplicity of element  $a$  in the multiset  $M$ . If  $U$  is finite,  $U = \{a_1, a_2, \dots, a_n\}$ , then  $M$  can also be represented by the vector  $(M(a_1), M(a_2), \dots, M(a_n))$  or by a string  $w = a_1^{M(a_1)} a_2^{M(a_2)} \dots a_n^{M(a_n)}$  (and all permutations of this string) where  $a^j$  denotes the string obtained by concatenating  $j \in \mathbb{N}$  occurrences of the letter  $a \in V$  (with  $a^0 = \lambda$ ). Note that the vector representation of a multiset coincides with the Parikh vector of the string representations.

We will also need the notion of a *finite transition system* which is defined as a triple  $M = (Q, q, \delta)$  where  $Q$  is a finite set of states,  $q \in Q$  is the initial state, and  $\delta : Q \rightarrow 2^Q$  is the state transition mapping. A state  $q' \in \delta(q)$  is called the successor state of  $q$ , and  $q \in Q$  is called a *halting state* if  $\delta(q) = \emptyset$ .

Lindenmayer systems (or *L systems*) are parallel rewriting systems introduced in [12] with the aim of describing the development of simple biological organisms in terms of formal languages. For more information on the area of L systems, see the book [13]. In the following, we will use the variants which are extended, tabled, and interactionless, that is, ETOL systems in short.

An *ETOL system* is a construct  $G = (V, T, U, w)$  where  $V$  is an alphabet,  $T \subseteq V$  is a terminal alphabet,  $w \in V^+$  is the initial word of  $G$ , and  $U = (P_1, \dots, P_m)$  where  $P_i$ ,  $1 \leq i \leq m$ , are finite sets of context-free productions over  $V$  (called *tables*), such that for each  $a \in V$ , there is at least one rule  $a \rightarrow \alpha$ ,  $\alpha \in V^*$  in each table. In a computational step in  $G$ , all the symbols of the current sentential form are rewritten using one of the tables of  $U$ . The language generated by  $G$  consists of all terminal strings which can be generated in a series of computational steps (a derivation) starting from the initial word, that is,  $L(G) = \{u \in T^* \mid w \Rightarrow^* u\}$  where  $\Rightarrow$  denotes a computational step, and  $\Rightarrow^*$  is the reflexive and transitive closure of  $\Rightarrow$ .

It is known (see [14], for example) that for each ETOL system with an arbitrary number of tables, there exists an ETOL system with only two tables generating the very same language.

Moreover, since we are going to relate ETOL languages to the multiset languages of P systems, we are not interested in the string generated by the ETOL system as a sequence of letters, but only in the multiplicities of different letters, that is, in the Parikh vectors of the generated strings. We will denote by  $Ps(G)$  the set of Parikh vectors corresponding to the strings of  $L(G)$  (also called the Parikh set of  $L(G)$ ), and by  $PsETOL$  the class of Parikh sets corresponding to the class of languages generated by ETOL systems.

Polymorphic membrane systems were introduced in [7]. Unlike in traditional membrane systems, the rules rules in polymorphic P systems are not fixed in advance, but they

are defined by the contents of specific membrane regions corresponding to the left- and right-hand sides of the rule.

A *polymorphic P system* of degree  $2n + 1$  for some  $n \in \mathbb{N}$  is a construct

$$\Pi = (O, T, \mu, w_s, \langle w_{1L}, w_{1R} \rangle, \dots, \langle w_{nL}, w_{nR} \rangle, h_o),$$

where  $O$  is the alphabet of objects,  $T \subseteq O$  is the set of terminal objects,  $\mu$  is the membrane structure consisting of  $2n + 1$  membranes labeled by a symbol from the set  $H = \{s, 1L, 1R, \dots, nL, nR\}$ , the elements of the multiset  $w_s$  are the initial contents of the skin membrane, the pairs of multi-sets  $\langle w_{iL}, w_{iR} \rangle$  correspond to the initial contents of membranes  $iL$  and  $iR$ ,  $1 \leq i \leq n$ , and  $h_o \in H$  is the label of the output membrane.

The membrane structure is usually denoted by a string of labeled and matching parentheses, but it can also be represented by a tree with its root labeled by the label of the outermost membrane, and the descendant nodes of each node labeled by the labels of membranes enclosed by the region corresponding to the given node. In the following, the number of nodes encountered during the traversal of the longest path from the root to a leaf in such a tree representation will be called the *depth* of the membrane system. (For example, the system which only has one membrane is of depth 1, while the system with a pair of nested membranes is of depth 2.) Note that for every  $1 \leq i \leq n$ , the membranes  $iL$  and  $iR$  have the same parent membrane, so they are located at the same depth.

The rules of  $\Pi$  are not given statically in the description, but are dynamically deduced for each configuration based on the content of the membrane pairs  $iL$  and  $iR$ ,  $1 \leq i \leq n$ . Thus, if in the configuration of the system these membranes contain the multisets  $u$  and  $v$ , then in the next step their parent membrane is transformed as if the  $u \rightarrow v$  multiset rewriting rule were added to it.

If there is at least one rule in a system  $\Pi$  where the number of objects in  $u$  (the multiset on the left-hand side) can grow to be greater than one, then we say that  $\Pi$  is a *cooperative* system, otherwise, it is a *non-cooperative* system. If  $iL$  is empty for some  $1 \leq i \leq n$  in a configuration, then the rule defined by the pair  $iL, iR$  is considered *disabled*, that is, no rule will be inferred from the contents of  $iL$  and  $iR$  for use in the next computational step.

A computation of the system is a series of computational steps in which the rules associated to a given region are applied in a maximally parallel way, that is, as many rules have to be applied in parallel as possible (with the restriction that each object can be rewritten by at most one rule). A P system halts (reaches a halting configuration) when no more computational steps are possible, that is, when no rule can be applied in any of the regions.

The set of vectors generated by a P system  $\Pi$  with the terminal alphabet  $T \subseteq O$  is the set of vectors representing the multisets of the terminal objects appearing in the output region  $h_o$  in a halting configuration of  $\Pi$  which is reached by a computation starting in the initial configuration of the system.

We denote the classes of non-cooperative polymorphic membrane systems and the sets of vectors they generate as  $OP(\text{polym}, \text{ncoo})$  and  $\mathcal{L}(OP(\text{polym}, \text{ncoo}))$ , respectively (*polym* stands for polymorphism, and *ncoo* means that the system is non-cooperative). To emphasize limited depth, we may also write  $OP^k(\text{polym}, \text{ncoo})$  and  $\mathcal{L}(OP^k(\text{polym}, \text{ncoo}))$  where  $k$  denotes the depth of the systems under consideration.

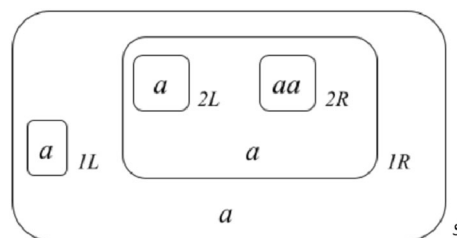
Now we recall an example of a simple polymorphic membrane system with super-exponential growth from [7].

**Example 1** Consider the polymorphic P system

$$\Pi_1 = (\{a\}, \{a\}, \mu, a, \langle a, a \rangle, \langle a, aa \rangle, s)$$

with membrane structure  $\mu = [ [ ]_{1L} [ [ ]_{2L} [ ]_{2R} ]_{1R} ]_s$ , as illustrated in Fig. 1.

In the initial configuration, the rule corresponding to the contents of  $1L, 1R$  (rule 1) is  $a \rightarrow a$ , and it will be applied in the skin region. The rule corresponding to  $2L, 2R$  (rule 2) is  $a \rightarrow aa$ , and it will be applied in region  $1R$ . In the first step, rule 1 is applied in the skin leaving the contents of the membrane intact, and rule 2 is applied in membrane  $1R$  doubling the number of  $a$ 's, so rule 1 (the rule corresponding to  $1L, 1R$ ) will be changed to  $a \rightarrow aa$ . In the second step, rule 1 will transform the multiset  $a$  in the skin into  $aa$ , and rule 2 will double the contents of region  $1R$  again, so after this step, rule 1 becomes  $a \rightarrow a^4$ . In general, after  $k$  derivation steps, the contents of  $1R$  will be  $a^{2^k}$ , so rule 1 will have the form  $a \rightarrow a^{2^k}$ . As the number of  $a$ 's in the skin will be  $2^{\frac{k(k-1)}{2}}$ , the rate of growth of the contents of the skin membrane is super-exponential.



**Fig. 1** The membrane structure and the initial configuration of the polymorphic P system  $\Pi_1$  of Example 1

### 3 Polymorphic P systems with limited depth

In this section, we examine the relationship of languages generated by ETOL systems and simple polymorphic P systems, where simplicity is captured by non-cooperation and limited depth. We look at an example first.

**Example 2** Consider the following ETOL system  $G = (V, T, U, w)$  with  $V = \{a_1, a_2\}, T = \{a_1\}, w = a_1a_2$ , and two tables  $U = (P_1, P_2)$ , each containing two rules

$$P_1 = \{a_1 \rightarrow a_1a_2, a_2 \rightarrow a_2a_1a_1\}, \text{ and}$$

$$P_2 = \{a_1 \rightarrow a_1, a_2 \rightarrow a_1\}.$$

We construct a non-cooperative polymorphic P system  $\Pi_2$  with depth 3 that can perform the choosing between rules of  $P_1$  and  $P_2$ , and therefore can simulate the operation of  $G$ .

Let  $O = \{a_1, a_2, a'_1, a'_2, a_1^0, a_2^0, a_1^1, a_2^1, \bar{a}_1, \bar{a}_2, \bar{\bar{a}}_1, \bar{\bar{a}}_2, b, c, d, F\}$ ,  $T' = \{a_1\}$  and

$$\Pi_2 = (O, T', \mu, w_s, \langle w_{1L}, w_{1R} \rangle, \dots, \langle w_{28L}, w_{28R} \rangle, s)$$

where the membrane structure of  $\Pi_2$  is such that the skin membrane directly contains the membranes  $1L, 1R, 17L, 17R, \dots, 28L, 28R$ , and the rest of the membranes are contained by  $1L$  and  $1R$ . In more detail,  $\mu$  is defined as

$$\mu = [ [\dots]_{1L} [\dots]_{1R} [ ]_{17L} [ ]_{17R} \dots [ ]_{28L} [ ]_{28R} ]_s,$$

where membrane  $1L$  contains the inner membranes  $[ ]_{2L} [ ]_{2R} \dots [ ]_{6L} [ ]_{6R}$ , and membrane  $1R$  contains the inner membranes  $[ ]_{7L} [ ]_{7R} \dots [ ]_{16L} [ ]_{16R}$ .

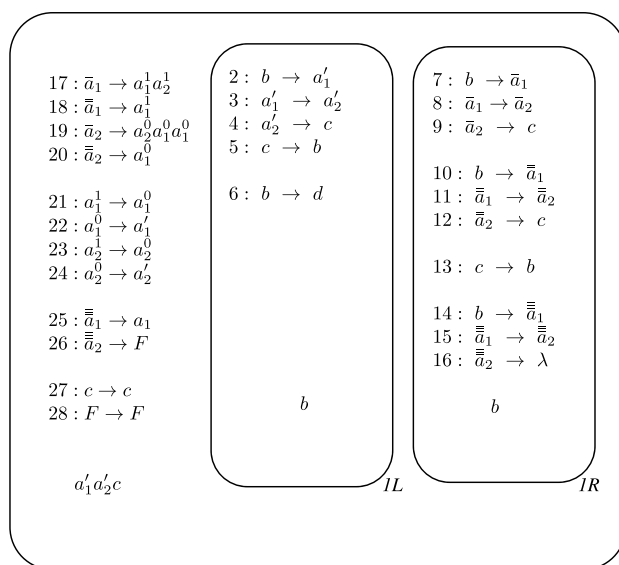
The graphical representation of  $\mu$  can be seen in Fig. 2 where also the initial membrane contents are depicted. Non-dynamical rules, that is, pairs of membranes  $[ w_{iL} ]_{iL}, [ w_{iR} ]_{iR}$  with constant contents (contents that never change during the computation) are given in a simplified notation as  $w_{iL} \rightarrow w_{iR}$ . Note that in this example, we only have one rule that changes dynamically, rule 1 (the rule corresponding to the regions  $1L, 1R$ ), the other rules have the same form at each step of the computation.

The initial contents of the regions with non-constant contents are

$$w_s = a'_1a'_2c, w_{1L} = b, w_{1R} = b,$$

the initial multisets contained by the rest of the regions are given (using the simplified notation) in Fig. 2.

The functioning of  $\Pi_2$  is demonstrated in Table 1. The first column contains the step number, the second column shows the form of rule 1 (defined by the membranes  $1L, 1R$ ) before the corresponding step, the third column contains the objects in the skin region, while the fourth, fifth, and



**Fig. 2** The membrane structure and the initial configuration of P system  $\Pi_2$  of Example 2. Non-dynamical rules are given in a simplified rewriting-rule-like notation

sixth columns contain the rules we (need to) use in the same computational steps.

The general idea behind the functioning of  $\Pi_2$  is as follows: Rules 17–20 simulate the rewriting process of the tables of  $G$ . Those with left-hand side  $\bar{a}_1$  or  $\bar{a}_2$  simulate the first table, those with left-hand side  $\bar{\bar{a}}_1, \bar{\bar{a}}_2$  simulate the second table. The objects of the skin region correspond to the sentential form of  $G$ . Rule 1 is “dynamic”, it prepares the objects of the skin membrane for the application of the rules 17–20 in the appropriate order. At the beginning of a “simulating cycle”, rule 1 is used to rewrite  $a_1$  (more precisely, its variant,  $a'_1$ ) to  $\bar{a}_1$  or  $\bar{\bar{a}}_1$  selecting this way the table to be simulated. Then, rule 1 changes to rewrite  $a'_2$  according to the same selection while rules 17–20 proceed with the actual simulation of the chosen table. The rest of the rules are needed to synchronize the whole process.

Table 1 shows how the rewriting of  $a_1a_2$  to  $a_1a_2a_2a_1a_1$  by the first table of  $G$  is simulated in  $\Pi_2$ . In the initial state, the form of rule 1 is  $b \rightarrow b$  which is not applicable because we only have objects  $a'_1, a'_2, c$  in the skin region, so we have to change rule 1 in the first step. In  $1L$  we can use rule 2 ( $b \rightarrow a'_1$ ) which rewrites  $b$  in  $1L$  to  $a'_1$  making rule 1 applicable. In parallel, we have to use rule 7 ( $b \rightarrow \bar{a}_1$ ) or rule 10 ( $b \rightarrow \bar{\bar{a}}_1$ ) in  $1R$  depending on the table of the ETOL system we want to simulate. To simulate  $P_1$ , we must use rule 7, to simulate  $P_2$ , we must use rule 10. As we would like to simulate  $P_1$ , we use rule 7. Rule 14 ( $b \rightarrow \bar{\bar{a}}_1$ ) is used only in the terminating phase of the computation, see below for more details.

As can be seen in the second row of Table 1, the form of rule 1 has changed, and now we can use it in the skin region

**Table 1** The first few computational steps of the polymorphic system  $\Pi_2$  of Example 2

Step	Rule 1	Contents of the Skin	Rule used in 1L	Rule used in 1R	Other rules used in the Skin
1	$b \rightarrow b$	$a'_1 a'_2 c$	2	7	27
2	$a'_1 \rightarrow \bar{a}_1$	$a'_1 a'_2 c$	3	8	27
3	$a'_2 \rightarrow \bar{a}_2$	$\bar{a}_1 a'_2 c$	4	9	17, 27
4	$c \rightarrow c$	$a_1^1 a_2^1 \bar{a}_2 c$	5	13	19, 21, 23, 27
5	$b \rightarrow b$	$a_1^0 a_2^0 a_1^0 a_1^0 c$	2	7 or 10	22, 24, 27
6	$a'_1 \rightarrow \bar{a}_1$ or $a'_1 \rightarrow \bar{\bar{a}}_1$	$a'_1 a'_2 a'_2 a'_1 a'_1 c$	.	.	.

to rewrite  $a'_1$  to  $\bar{a}_1$ . At the same time, the rules used in 1L and 1R ( $a'_1 \rightarrow a'_2, \bar{a}_1 \rightarrow \bar{a}_2$ , respectively) change the form of rule 1 to  $a'_2 \rightarrow \bar{a}_2$  in order to be able to start rewriting  $a'_2$ -s in the next step.

After we have used rule 1 and the objects in the skin region have changed, we can use rule 17 ( $\bar{a}_1 \rightarrow a_1^1 a_2^1$ ) which simulates the first rule from the table  $P_1$  of  $G$ . The upper indexing of the symbols on the right-hand side starts from 1, and it will decrease in each of the following steps until the symbols are written back into the original primed form (after counting down with the indices to zero) at the appropriate step, that is, at the step when the rewriting of the other symbol,  $a'_2$  is also finished.

Meanwhile, in step 3, rule 1 ( $a'_2 \rightarrow \bar{a}_2$ ) is also applied to rewrite  $a'_2$  (so the second rule of table  $P_1$  of  $G$  can also be simulated), and rule 1 is changed to  $c \rightarrow c$  (so it does not change the contents of the skin region in the next step).

Then, with rule 19 ( $\bar{a}_2 \rightarrow a_2^0 a_1^0 a_1^0$ ), the rule  $a_2 \rightarrow a_2 a_1 a_1$  (the second rule of  $P_1$ ) is simulated, while rules 21 and 23 decrement the upper indices of the objects introduced by the simulation of the previous rule, and the form of rule 1 is changed to  $b \rightarrow b$ .

Now, as can be seen in row 5 of Table 1, the system is ready to prepare the next simulating cycle by rewriting the objects corresponding to the sentential form of  $G$  to their original primed versions, and changing rule 1 in the appropriate way. We can return to a state that is similar to the initial state by choosing between rule 7 and rule 10 again (to simulate another step from the ETOL system), and in parallel, by rewriting  $a_1^0$ -s and  $a_2^0$ -s to  $a'_1$ -s and  $a'_2$ -s with rules 22 and 24.

The simulation of the ETOL system can be stopped at the steps which precede the table selection phase of the simulation. If both 1L and 1R contain the object  $b$  and we want to stop the simulating process, we can choose rule 14 ( $b \rightarrow \bar{\bar{a}}_1$ ) instead of rule 7 or 10, which will change the symbols to their triple-barred versions in the skin membrane. As  $a_1 \in T$ , the symbol  $\bar{\bar{a}}_1$  can be rewritten to the terminal symbol  $a_1$ , but since  $a_2 \notin T$ , the symbol  $\bar{\bar{a}}_2$  is changed to  $F$  which will never result in a halting configuration (because of the rule  $F \rightarrow F$ ). In the next step, rule 4 ( $a'_2 \rightarrow c$ ) and

rule 16 ( $\bar{\bar{a}}_2 \rightarrow \lambda$ ) is used in the regions 1L and 1R, respectively, so it will be possible to remove the symbol  $c$  from the skin, enabling the halting of the system in the case when no nonterminal symbols are present. Otherwise, if nonterminal objects were present in the beginning of the finishing phase, then the symbol  $F$  was introduced, so rule 27 ( $F \rightarrow F$ ) prevents the halting of the system. The finishing phase of the computation is shown in Table 2.

The halting of the system is also impossible if rule 6 ( $b \rightarrow d$ ) is used "too early" (when rules 14 – 16 were not yet applied in region 1R) since the symbol  $c$  and the rule  $c \rightarrow c$  is present in the skin region. Thus, the result of a halting computation of  $\Pi_2$  is a multiset over  $T = \{a_1\}$  in the skin membrane (the output membrane of the system) which corresponds to the Parikh set of a string that can be generated by the ETOL system  $G$ .

Now we show how the idea presented in the example above can be generalized to arbitrary ETOL systems.

**Theorem 1**  $PsETOL \subseteq \mathcal{L}(OP^3(polym, ncoo))$ .

**Proof** Let  $G = (V, T, U, w)$  be an ETOL system, let  $k$  denote the number of letters in the alphabet,  $V = \{a_1, a_2, \dots, a_k\}$ , and let  $T = \{a_1, \dots, a_l\} \subseteq V$  for some  $l \leq k$ . Without loss of generality, we assume that  $G$  has exactly two tables,  $U = (P_1, P_2)$ . We denote the  $j$ th rule of table  $i$ ,  $1 \leq i \leq 2$

**Table 2** The last computational steps of a successful computation in the polymorphic system  $\Pi_2$  of Example 2

Step	Rule 1	Contents of the Skin	Rule used in 1L	Rule used in 1R	Other rules used in the Skin
$(n - 4)$	$b \rightarrow b$	$a_1^0 \dots a_1^0 c$	2	14	22, 27
$(n - 3)$	$a'_1 \rightarrow \bar{\bar{a}}_1$	$a'_1 \dots a'_1 c$	3	15	27
$(n - 2)$	$a'_2 \rightarrow \bar{\bar{a}}_2$	$\bar{\bar{a}}_1 \dots \bar{\bar{a}}_1 c$	4	16	25, 27
$(n - 1)$	$c \rightarrow \lambda$	$a_1 \dots a_1 c$	5	–	–
$(n)$	$b \rightarrow \lambda$	$a_1 \dots a_1$	6	–	–
	$d \rightarrow \lambda$	$a_1 \dots a_1$	–	–	–

(and their left- and right-hand sides) as  $\alpha_{i,j} \rightarrow \beta_{i,j}$ , where  $\alpha_{i,j} \in V$  and  $\beta_{i,j} \in V^*$ ,  $1 \leq j \leq m$ . In order to simplify the notation, we assume that the cardinality of the two tables are the same,  $m = |P_1| = |P_2|$ . If this is not the case, that is, if  $|P_1| < |P_2| = m$ , then  $\alpha_{1,j} \rightarrow \beta_{1,j}$  for  $|P_1| + 1 \leq j \leq |P_2|$  (or  $\alpha_{2,j} \rightarrow \beta_{2,j}$  for  $|P_2| + 1 \leq j \leq |P_1|$  if  $|P_2| < |P_1| = m$ ) will denote the same rule as  $\alpha_{1,1} \rightarrow \beta_{1,1}$  (or  $\alpha_{2,1} \rightarrow \beta_{2,1}$ ).

Let

$$O = \{a'_i, a''_i, \bar{a}_i, \bar{\bar{a}}_i, \bar{\bar{\bar{a}}}_i \mid 1 \leq i, n \leq k\} \cup \{a_i \mid 1 \leq i \leq l\} \cup \{b, c, d, F\},$$

$$T = \{a_i \mid 1 \leq i \leq l\},$$

and let

$$\Pi = (O, T, \mu, w_s, \langle w_{1L}, w_{1R} \rangle, \dots, \langle w_{pL}, w_{pR} \rangle, s)$$

where  $p = (5k + 8) + 2m + k^2 + 2$  and the membrane structure of  $\Pi$  is such that the skin region directly contains the membranes with labels  $1L, 1R$  and  $iL, iR$  for  $(4k + 9) \leq i \leq p$ , while the rest of the membranes are contained by  $1L$  and  $1R$ . In more detail,  $\mu$  is defined as

$$\mu = [ [\dots]_{1L} [\dots]_{1R} [ ]_{(4k+9)L} [ ]_{(4k+9)R} \dots [ ]_{pL} [ ]_{pR} ]_s$$

with membrane  $1L$  containing the membranes  $[ ]_{2L} [ ]_{2R} \dots [ ]_{(k+4)L} [ ]_{(k+4)R}$ , and membrane  $1R$  containing the membranes  $[ ]_{(k+5)L} [ ]_{(k+5)R} \dots [ ]_{(4k+8)L} [ ]_{(4k+8)R}$ .

In  $1L$ , the number of rules depends on the number of letters in the alphabet of the ETOL system, we have to apply  $k + 2$  rules for each table simulation in succession (where  $k = |V|$ ). In general, we specify the rules for the  $k$  letters as

$$r_2 : b \rightarrow a'_1, r_{i+2} : a'_i \rightarrow a'_{i+1}, \text{ for } 1 \leq i \leq k - 1, \text{ and}$$

$$r_{k+2} : a'_k \rightarrow c, r_{k+3} : c \rightarrow b.$$

These rules perform the same task as the rules of  $1L$  in Example 2 do for two letters. In order to be able to finish the simulation, we also need the additional rule  $r_{k+4} : b \rightarrow d$  (see the end of the proof for more details).

Note that here we have used the simplified notation again for membranes with contents that remain constant for the whole computation. (Without this simplification we would have to write  $\langle w_{2L}, w_{2R} \rangle$  and specify  $w_{2L} = b, w_{2R} = a'_1$  instead of the rule  $r_2 : b \rightarrow a'_1$ , for example.)

Rules applied in  $1R$  determine on the choice of the table, so we have to introduce rules for  $P_1$  and  $P_2$ . We need

$$r_{k+5} : b \rightarrow \bar{a}_1, r_{k+5+i} : \bar{a}_i \rightarrow \bar{a}_{i+1} \text{ for } 1 \leq i \leq k - 1,$$

$$r_{2k+5} : \bar{a}_k \rightarrow c,$$

to enable the simulation of  $P_1$ , the rules

$$r_{2k+6} : b \rightarrow \bar{\bar{a}}_1, r_{2k+6+i} : \bar{\bar{a}}_i \rightarrow \bar{\bar{a}}_{i+1} \text{ for } 1 \leq i \leq k - 1,$$

$$r_{3k+6} : \bar{\bar{a}}_k \rightarrow c,$$

to enable the simulation of  $P_2$ , and one more rule  $r_{3k+7} : c \rightarrow b$  to finish the process.

To end the simulation, we need to be able to stop the functioning of  $\Pi$  in the case when the skin region contains (primed variants of) terminal letters only, that is, objects only from the set  $\{a'_1, \dots, a'_k\} = \{a' \mid a \in T\}$ . In order to do this, we introduce a mechanism that is similar to the simulation of the tables. To  $1R$ , we add the rules

$$r_{3k+8} : b \rightarrow \bar{\bar{\bar{a}}}_1, r_{3k+8+i} : \bar{\bar{\bar{a}}}_i \rightarrow \bar{\bar{\bar{a}}}_{i+1} \text{ for}$$

$$1 \leq i \leq k - 1, r_{4k+8} : \bar{\bar{\bar{a}}}_k \rightarrow \lambda.$$

In the skin region, we go through the objects of the alphabet applying the rules of the chosen table to each of them in a sequence, the rules for the occurrences of one specific object at a time. For this reason, we use indexed variants of the symbols which are produced in the intermediate steps, so they are able to “wait” until the rules for the rest of the objects are also applied. To achieve this effect, we add extra rules decreasing the indices in such a way that we get back to the primed form  $a'_1, a'_2, \dots, a'_k$  for all objects at the same computational step.

Recall our assumption that  $|P_1| = |P_2| = m$ , and that we denote the  $j$ th rule of table  $i$ ,  $1 \leq i \leq 2$ , as  $\alpha_{i,j} \rightarrow \beta_{i,j}$  where  $\alpha_{i,j} \in V$  and  $\beta_{i,j} \in V^*$ ,  $1 \leq j \leq m$ . In the following we will use the notation  $\beta^l_{i,j}$  for some  $1 \leq l \leq k$  to express that  $\beta^l_{i,j} \in \{a'_1, a'_2, \dots, a'_k\}^*$  is the indexed version of the corresponding string  $\beta_{i,j} \in \{a_1, a_2, \dots, a_k\}^*$ .

Now we add the following rules to the skin region. To simulate the rules of  $P_1$ , we need

$$r_{4k+8+j} : \bar{a}_i \rightarrow \beta^{k+1-i}_{1,j} \text{ for each rule } \alpha_{1,j} \rightarrow \beta_{1,j} \in P_1$$

where  $\alpha_{1,j} = a_i$  for some  $a_i \in \{a_1, a_2, \dots, a_k\}$  and  $1 \leq j \leq m$ .

For the simulation of  $P_2$ , we have

$$r_{4k+8+m+j} : \bar{\bar{a}}_i \rightarrow \beta^{k+1-i}_{2,j} \text{ for each rule } \alpha_{2,j} \rightarrow \beta_{2,j} \in P_2$$

where  $\alpha_{2,j} = a_i$  for some  $a_i \in \{a_1, a_2, \dots, a_k\}$  and  $1 \leq j \leq m$ .

After rewriting with the rules above, we have to use rules to count down with the indices of the objects that were produced until the last element of the alphabet is rewritten (similarly to the way we count down in the example). To achieve this, for each  $a_i$ ,  $1 \leq i \leq k$ , we need

$$r_{4k+8+2m+(i-1)k+n} : a^n_i \rightarrow a^{n-1}_i \text{ for } 2 \leq n \leq k, \text{ and}$$

$$r_{4k+8+2m+(i-1)k+1} : a^1_i \rightarrow a'_i.$$

In order to stop the system, we use the rules

$$r_{4k+8+2m+k^2+i} : \bar{\bar{\bar{\delta}}}_i \rightarrow \delta_i \text{ where } \delta_i = a_i \text{ if } a_i \in T \text{ or}$$

$$\delta_i = F \text{ if } a_i \notin T,$$

for  $1 \leq i \leq k$  in the skin region. If the multiset in the skin region corresponding to the current sentential form of the simulated ETOL system contains primed versions of terminals only, that is, elements of the set  $\{a'_1, \dots, a'_l\}$ , then these rules rewrite them to terminal objects. Otherwise, if nonterminal objects are present (that is, if the simulation should not be stopped), they introduce the nonterminal  $F$  which prevents the halting of the system because the rule

$$r_{5k+8+2m+k^2+1} : F \rightarrow F$$

is also present in the skin region.

The simulation of the ETOL system can be stopped as follows: If both  $1L$  and  $1R$  contain the object  $b$ , then choosing rule  $r_{3k+8} : b \rightarrow \bar{\bar{a}}$  (instead of rule  $r_{k+5} : b \rightarrow \bar{a}$  or  $r_{2k+6} : b \rightarrow \bar{a}$ ) will change the symbols to their triple-barred versions and then to terminals in the skin membrane, and produce  $c$  and  $\lambda$  in region  $1L$  and  $1R$ , respectively, so the object  $c$  can also be removed from the skin region. Now, if rule  $r_{k+4} : b \rightarrow d$  is chosen to be applied in region  $1L$ , then the system halts producing a multiset of terminals in the skin.

If nonterminal objects were still present in the skin region at the beginning of the finishing phase (that is, if the simulation should not produce any result), then the symbol  $F$  was introduced, so the rule  $r_{5k+8+2m+k^2+1} : F \rightarrow F$  prevents the halting of the system.

If rule  $r_{k+4} : b \rightarrow d$  is applied before the nonterminals are removed from the skin region, then the symbol  $c$  is still present, so if we also add the rule

$$r_{5k+8+2m+k^2+2} : c \rightarrow c,$$

it prevents the halting of the system.

When  $\Pi$  halts, the result is a multiset over  $T = \{a_1, a_2, \dots, a_l\}$  which corresponds to a terminal string that can be generated by the ETOL system  $G$ .

To see that the P system  $\Pi$  cannot produce multisets that do not correspond to the Parikh set of a string generated by the ETOL system, observe the construction of  $\Pi$ . Informally speaking, there are just a few points where the rule application in  $\Pi$  is not deterministic (apart from simulating the possibility when one symbol can be rewritten by more than one rule of the ETOL system). These nondeterministic steps involve the choice of the table of  $G$  to be simulated and the choice whether to finish or to continue the computation, so we might conclude that the behavior of the P system corresponds to an ETOL system derivation.  $\square$

## 4 Polymorphic P systems with finite sets of instances of dynamic rules

Since there is no communication between the regions (as we consider P systems with “no ingredients”), the sequence of multisets appearing inside a given region as the computation proceeds only depends on the initial contents of the region itself. To formalize this idea, we introduce a successor relation defined on the contents of regions.

Let  $\Pi = (O, T, \mu, w_s, \langle w_{1L}, w_{1R} \rangle, \dots, \langle w_{nL}, w_{nR} \rangle, h_o)$  be a polymorphic P system, and let  $w_{j,h}$  for some  $h \in \{s, 1L, 1R, \dots, nL, nR\}$  denote the multiset contained by the region labeled by  $h$  after the  $j$ th step of the computation of  $\Pi$  for some  $j \geq 0$ . We say that  $w'_{j,h}$  is in the *successor set* of  $w_{j,h}$ , denoted as  $w'_{j,h} \in \sigma_{j,h}(w_{j,h})$ , if  $w'_{j,h}$  can be obtained from  $w_{j,h}$  by the maximally parallel applications of the multiset rewriting rules associated to the region  $h$ , as can be deduced from the configuration of  $\Pi$  which is reached in the  $j$ th step of the computation.

If for the same  $w_{j,h}$  as above, we fix  $\sigma_{j,h}^0(w_{j,h}) = \{w_{j,h}\}$  for any  $j \geq 0$ , and for  $k \geq 0$  we have  $\sigma_{j,h}^{k+1} = \sigma_{j+k,h}(\sigma_{j,h}^k(w_{j,h}))$  (where we extend the range of the function from multisets to sets of multisets in the natural way), then we can define

$$\sigma_{j,h}^* = \bigcup_{k \geq 0} \sigma_{j,h}^k(w_{j,h}).$$

**Definition 1** Given a polymorphic P system  $\Pi$  as above, we say that a region  $h$  of  $\Pi$  is *finitely representable* or *FIN-representable* in short, if the set of successor multisets of the initial contents of  $h$ ,  $w_h$ , is finite, that is,  $\sigma_{0,h}^*(w_h)$  is finite.

First, we demonstrate the notion of FIN-representability on an example.

**Example 3** Consider the polymorphic P system

$$\Pi_3 = (O, T, \mu, w_s, \langle w_{1L}, w_{1R} \rangle, \dots, \langle w_{8L}, w_{8R} \rangle, s)$$

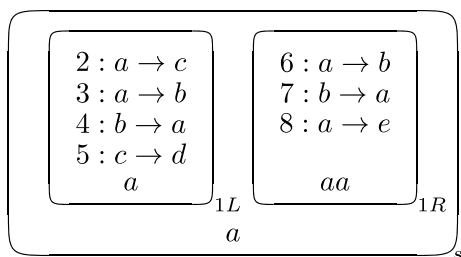
where  $O = T = \{a, b, c, d, e\}$ , and the membrane structure is  $\mu = [ [\dots]_{1L} [\dots]_{1R} ]_s$ , where the child membranes of  $1L$  are  $[ ]_{2L} [ ]_{2R} \dots [ ]_{5L} [ ]_{5R}$  and the children of  $1R$  are  $[ ]_{6L} [ ]_{6R} \dots [ ]_{8L} [ ]_{8R}$ . Let

$$w_s = a, w_{1L} = a, w_{1R} = aa,$$

and using the simplified notation for static (non-dynamic) rules, let the rules applicable in  $1L$  be

$$r_2 : a \rightarrow c, r_3 : a \rightarrow b, r_4 : b \rightarrow a, r_5 : c \rightarrow d,$$

and the rules applicable in  $1R$  be



**Fig. 3** The initial configuration of the polymorphic P system  $\Pi_3$  of Example 3

$$r_6 : a \rightarrow b, r_7 : b \rightarrow a, r_8 : a \rightarrow e,$$

as can also be seen in Fig. 3.

In the following, we show that both 1L and 1R are FIN-representable. Concerning 1L, observe that  $\sigma_{0,1L}^*(a) = \{a, b, c, d\}$  with  $\sigma_{0,1L}(a) = \sigma_{j,1L}(a) = \{b, c\}$ ,  $\sigma_{0,1L}(b) = \sigma_{j,1L}(b) = \{a\}$ ,  $\sigma_{0,1L}(c) = \sigma_{j,1L}(c) = \{d\}$ , and  $\sigma_{0,1L}(d) = \sigma_{j,1L}(d) = \emptyset$  for all  $j \geq 0$ , which can be represented with the graph in Fig. 4a.

Considering 1R,  $\sigma_{0,1R}(aa) = \sigma_{j,1R}(aa) = \{bb, be, ee\}$ ,  $\sigma_{0,1R}(ae) = \sigma_{j,1R}(ae) = \{be, ee\}$ ,  $\sigma_{0,1R}(bb) = \sigma_{j,1R}(bb) = \{aa\}$ ,  $\sigma_{0,1R}(be) = \sigma_{j,1R}(be) = \{ae\}$ , and  $\sigma_{0,1R}(ee) = \sigma_{j,1R}(ee) = \emptyset$ , thus,  $\sigma_{0,1R}^*(aa) = \{aa, bb, ae, be, ee\}$ , as can be seen on Fig. 4b.

The skin region is not FIN-representable, as the dynamical rule  $r_1$  given by the membranes labeled with 1L and 1R has more than one symbol on its right-hand side in each computational step, and this means that the number of symbols in the skin region is increasing with each rule application. To see this, consider the following. Initially,  $r_1 : a \rightarrow aa$ , so  $\sigma_{0,s}(a) = \{aa\}$ . Then  $r_1 \in \{c \rightarrow be, c \rightarrow ee, c \rightarrow bb, b \rightarrow be, b \rightarrow ee, b \rightarrow bb\}$ , so it is not applicable,  $\sigma_{0,s}^2(a) = \sigma_{1,s}(aa) = \{aa\}$ . Then  $r_1 \in \{d \rightarrow aa, d \rightarrow ae, d \rightarrow ee, a \rightarrow aa, a \rightarrow ae, a \rightarrow ee\}$ , thus,  $\sigma_{0,s}^3(a) = \sigma_{2,s}^2(aa) = \sigma_{3,s}(aaa) = \{aaaa, aaae, aae, ae, ee, eeee\}$ . As we see by the construction of the system, the rule  $a \rightarrow aa$  might appear arbitrary many times during the subsequent parts of the computation, the multiplicity of objects in the

skin region might keep increasing, so this region cannot be FIN-representable according to Definition 1.

Given a non-cooperative polymorphic system as defined above, left-hand sides of rules have at most one symbol, so the membranes with labels  $iL$ ,  $1 \leq i \leq n$ , are always FIN-representable. The situation is different, however, in the case of membranes  $iR$ ,  $1 \leq i \leq n$ . If at least one of the rules which is applicable (in an arbitrary number of computational steps) during a computation inside some right-hand membrane has more than one symbol on its right-hand side, then the corresponding dynamic rule has arbitrary many possible instances, so the region corresponding to its right-hand side cannot be FIN-representable.

Let us denote by  $OP(polym, ncoo, fin)$  the class of non-cooperative polymorphic membrane systems of degree  $2n + 1$  where all right-hand side regions,  $iR$ ,  $1 \leq i \leq n$ , are FIN-representable, and let  $\mathcal{L}(OP(polym, ncoo, fin))$  denote the class of vectors that they generate.

If a region is FIN-representable, we can describe the sequences of possible contents that can appear during the computations with a finite transition system.

**Definition 2** Let  $\Pi \in OP(polym, ncoo, fin)$  be a FIN-representable polymorphic P system with  $\Pi = (O, T, \mu, w_s, \langle w_{1L}, w_{1R} \rangle, \dots, \langle w_{nL}, w_{nR} \rangle, s)$ , and let  $M = (Q, q_0, \delta)$  be a transition system. We say that  $M$  represents the rule configurations of  $\Pi$  if the contents of the pairs of membranes labeled by 1L, 1R, ..., nL, nR of the configuration sequences of  $\Pi$

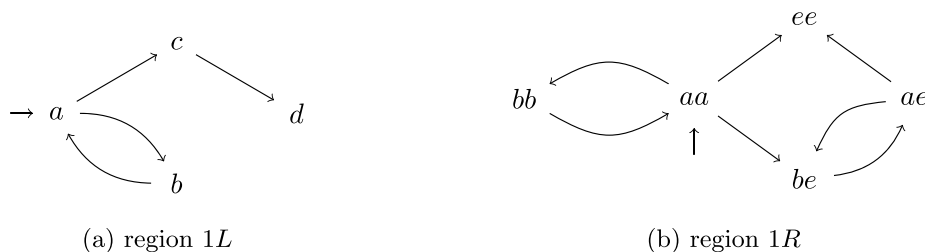
$$\begin{aligned} & (w_s, w_{1L}, w_{1R}, \dots, w_{nL}, w_{nR}) \\ &= (w_s^0, w_{1L}^0, w_{1R}^0, \dots, w_{nL}^0, w_{nR}^0) \Rightarrow \\ &\Rightarrow (w_s^1, w_{1L}^1, w_{1R}^1, \dots, w_{nL}^1, w_{nR}^1) \Rightarrow \dots \Rightarrow \\ &(w_s^i, w_{1L}^i, w_{1R}^i, \dots, w_{nL}^i, w_{nR}^i) \Rightarrow \dots \end{aligned}$$

are in a one-to-one correspondence with the state sequences

$$q_0 = q^0 \Rightarrow q^1 \Rightarrow \dots \Rightarrow q^i \Rightarrow \dots$$

of the transition system  $M$ . Moreover,  $q^h$ ,  $h \geq 0$ , is a halting state of  $M$ , if and only if no rules are applicable in any of the regions 1L, 1R, ..., nL, nR in the corresponding configuration

**Fig. 4** Graphical demonstration of the possible membrane contents of the P system  $\Pi_3$  of Example 3 with arrows indicating the initial contents and the successor relation between the multisets (which are given by their string representations)





$(w_s^h, w_{1L}^h, w_{1R}^h, \dots, w_{nL}^h, w_{nR}^h)$  of  $\Pi$ . Note that  $M$ , as the evolution of the rule configurations of  $\Pi$  can be nondeterministic.

In the following lemma, we describe how such a representation can be constructed.

**Lemma 2** *For any polymorphic P system  $\Pi \in OP(polym, ncoo, fin)$ , we can construct a finite transition system  $M_\Pi$  which represents the rule configurations of  $\Pi$ .*

**Proof** Let  $\Pi \in OP(polym, ncoo, fin)$  be a polymorphic P system with  $\Pi = (O, T, \mu, w_s, \langle w_{1L}, w_{1R} \rangle, \dots, \langle w_{nL}, w_{nR} \rangle, s)$ , and let  $1L, 1R, \dots, lL, lR$  for some  $l \leq n$  be the labels of those regions which are directly enclosed in the skin membrane. We construct a finite transition system  $M_{1\dots l}$  which represents the rule configurations of  $\Pi$  by describing how the  $k$ -tuples of rules corresponding to these regions are changing during the computations.

We start with constructing representations of the elementary membranes at the deepest level of the system, and continue upwards, level by level, finally arriving to those regions which are enclosed by the skin membrane.

(1) If a membrane labeled by  $h \in \{1L, 1R, \dots, nL, nR\}$  is an elementary membrane, then let  $M_h = (Q_h, \bar{q}_h, \delta_h)$ , where

- $Q_h = \{\bar{q}_h\} = \{(w_h, \emptyset)\}$ , and
- $\delta_h : Q_h \rightarrow 2^{Q_h}$  such that  $\delta_h(\bar{q}_h) = \emptyset$ .

This transition system has a single state which is a pair of the unchanging membrane contents and an empty set, the set of rules that can be used for changing the contents.

As we move upwards in the hierarchy, a similar representation will be constructed for each region: the states of the transition systems are pairs of the (dynamically changing) membrane contents and sets of rules which can be used to transform them.

(2) If we have already constructed  $M_{iL} = (Q_{iL}, \bar{q}_{iL}, \delta_{iL})$  and  $M_{iR} = (Q_{iR}, \bar{q}_{iR}, \delta_{iR})$  for the pair of elementary membranes labeled by  $iL, iR$  for some  $1 \leq i \leq n$ , we construct  $M_i = (R_i, \bar{r}_i, \delta_i)$  to represent the (non-dynamical) rule corresponding to this pair of membranes with

- $R_i = \{\bar{r}_i\} = \{(\bar{q}_{iL}, \bar{q}_{iR})\}$  where  $\bar{q}_{iL} = (w_{iL}, \emptyset), \bar{q}_{iR} = (w_{iR}, \emptyset)$ , and
- $\delta_i : R_i \rightarrow 2^{R_i}$  such that  $\delta_i(\bar{r}_i) = \emptyset$ .

We also say that (using the above notation)  $\bar{r}_i$  corresponds to the rule  $w_{iL} \rightarrow w_{iR}$ .

(3) If the elementary membranes that are directly enclosed by the non-elementary membrane (their parent membrane) with label  $h$  are labeled by  $i_1L, i_1R, \dots, i_kL, i_kR$ ,

and we have already constructed  $M_{i_1}, \dots, M_{i_k}$  with  $M_{i_j} = (R_{i_j}, \bar{r}_{i_j}, \delta_{i_j})$  for all the pairs  $i_jL, i_jR, 1 \leq j \leq k$ , then we construct the representation  $M_h$  in two steps.

(3.1) We first construct  $M_{i_1\dots i_k} = (R_{i_1\dots i_k}, \bar{r}_{i_1\dots i_k}, \delta_{i_1\dots i_k})$  with

- $R_{i_1\dots i_k} = \{\bar{r}_{i_1\dots i_k}\} = \{(\bar{r}_{i_1}, \dots, \bar{r}_{i_k})\}$ , and
- $\delta_{i_1\dots i_k} : R_{i_1\dots i_k} \rightarrow 2^{R_{i_1\dots i_k}}$  such that  $\delta_{i_1\dots i_k}(r_{i_1\dots i_k}) = \emptyset$ .

We say that  $\bar{r}_{i_1\dots i_k}$  corresponds to the set of rules  $\{w_{i_1L} \rightarrow w_{i_1R}, \dots, w_{i_kL} \rightarrow w_{i_kR}\}$ .

(3.2) Given  $M_{i_1\dots i_k}$  and the initial multiset  $w_h$ , we can construct  $M_h = (Q_h, \bar{q}_h, \delta_h)$  as

- $Q_h = \sigma_{0,h}^*(w_h) \times \{\bar{r}_{i_1\dots i_k}\}$ , the direct product of the possible contents of region  $h$  and the representation of the set of rules corresponding the pairs of (elementary) regions  $i_jL, i_jR, 1 \leq j \leq k$ ,
- $\bar{q}_h = (w_h, \bar{r}_{i_1\dots i_k})$ , the pair of the initial contents and representation of the set of rules associated to the region, and
- $\delta_h : Q_h \rightarrow 2^{Q_h}$  such that  $(u'_h, \bar{r}_{i_1\dots i_k}) \in \delta_h(u_h, \bar{r}_{i_1\dots i_k})$  if and only if
  - the multiset  $u'_h$  can be obtained from  $u_h$  by the maximally parallel application of the set of rules  $\{r_{i_1}, \dots, r_{i_k}\}$  corresponding to  $\bar{r}_{i_1\dots i_k}$ , denoted as  $u_h \Rightarrow_{\{r_{i_1}, \dots, r_{i_k}\}} u'_h$ , or
  - if none of the rules corresponding to  $\bar{r}_{i_1\dots i_k}$  are applicable to  $u_h$ , then the state  $(u_h, \bar{r}_{i_1\dots i_k})$  is a halting state in  $M_h, \delta_h(u_h, \bar{r}_{i_1\dots i_k}) = \emptyset$ .

Now we repeat the above procedure also for the non-elementary membranes as follows:

(4) If we have already constructed  $M_{iL} = (Q_{iL}, \bar{q}_{iL}, \delta_{iL})$  and  $M_{iR} = (Q_{iR}, \bar{q}_{iR}, \delta_{iR})$  for the pair of (not necessarily elementary) membranes labeled by  $iL, iR$  for some  $1 \leq i \leq n$ , we construct  $M_i = (R_i, \bar{r}_i, \delta_i)$  to represent the dynamical rule corresponding to this pair of membranes. Let

- $R_i = Q_{iL} \times Q_{iR}$ ,
- $\bar{r}_i = (\bar{q}_{iL}, \bar{q}_{iR})$ , and
- $\delta_i : R_i \rightarrow 2^{R_i}$  such that  $(q'_{iL}, q'_{iR}) \in \delta_i(q_{iL}, q_{iR})$ , if and only if
  - $q'_{iL} \in \delta_{iL}(q_{iL})$  and  $q'_{iR} \in \delta_{iR}(q_{iR})$ , or
  - if  $\delta_{iL}(q_{iL}) = \emptyset$  and  $q'_{iR} \in \delta_{iR}(q_{iR})$  (or if  $\delta_{iR}(q_{iR}) = \emptyset$  and  $q'_{iL} \in \delta_{iL}(q_{iL})$ ) then  $q'_{iL} = q_{iL}$  (or  $q'_{iR} = q_{iR}$ , respectively), or

$\delta_i(q_{iL}, q_{iR}) = \emptyset$ , if and only if and  $\delta_{iL}(q_{iL}) = \delta_{iR}(q_{iR}) = \emptyset$ .

(5) If the membranes that are directly enclosed by the non-elementary membrane (their parent membrane) with label  $h$  are labeled by  $i_1L, i_1R, \dots, i_kL, i_kR$ , and we have already constructed  $M_{i_1}, \dots, M_{i_k}$   $M_{i_j} = (R_{i_j}, \bar{r}_{i_j}, \delta_{i_j})$  for all the pairs  $i_jL, i_jR, 1 \leq j \leq k$ , then we construct the representation  $M_h$  in two steps.

(5.1) We first construct  $M_{i_1 \dots i_k} = (R_{i_1 \dots i_k}, \bar{r}_{i_1 \dots i_k}, \delta_{i_1 \dots i_k})$  with

- $R_{i_1 \dots i_k} = R_{i_1} \times \dots \times R_{i_k}$ ,
- $\bar{r}_{i_1 \dots i_k} = (\bar{r}_{i_1}, \dots, \bar{r}_{i_k})$ , and
- $\delta_{i_1 \dots i_k} : R_{i_1 \dots i_k} \rightarrow 2^{R_{i_1 \dots i_k}}$  such that  $(r'_1, \dots, r'_k) \in \delta_{i_1 \dots i_k}(r_1, \dots, r_k)$ , if and only if
  - $r'_j \in \delta_{i_j}(r_j)$  for all  $i_j, 1 \leq j \leq k$ , or
  - if  $\delta_{i_j}(r_j) = \emptyset$ , but there is at least one  $i_l$ , such that  $\delta_{i_l}(r_l) \neq \emptyset, 1 \leq l \leq k$ , then  $r'_j = r_j$ , and  $\delta_{i_1 \dots i_k}(r_1, \dots, r_k) = \emptyset$ , if and only if,  $\delta_{i_j}(r_j) = \emptyset$  for all  $i_j, 1 \leq j \leq k$ .

(5.2) Given  $M_{i_1 \dots i_k}$  and the initial multiset  $w_h$ , we can construct  $M_h = (Q_h, \bar{q}_h, \delta_h)$  as

- $Q_h = \sigma_{0,h}^*(w_h) \times R_{i_1 \dots i_k}$ , the direct product of the possible contents of region  $h$  and the representation of the possible  $k$  element sets of rules determined by the actual configuration of the pairs of regions  $i_jL, i_jR, 1 \leq j \leq k$ ,
- $\bar{q}_h = (w_h, \bar{r}_{i_1 \dots i_k})$ , the pair of the initial contents and the set of rules represented by the initial configuration, and
- $\delta_h : Q_h \rightarrow 2^{Q_h}$  such that  $(u'_h, r'_{i_1 \dots i_k}) \in \delta_h(u_h, r_{i_1 \dots i_k})$  if and only if
  - the multiset  $u'_h$  can be obtained from  $u_h$  by the maximally parallel application of the set of rules of  $\{r_{i_1}, \dots, r_{i_k}\}$  corresponding to  $r_{i_1 \dots i_k}$  denoted as  $u_h \Rightarrow_{\{r_{i_1}, \dots, r_{i_k}\}} u'_h$ , and
  - $r'_{i_1 \dots i_k} \in \delta_{i_1 \dots i_k}(r_{i_1 \dots i_k})$ , or if  $\delta_{i_1 \dots i_k}(r_{i_1 \dots i_k}) = \emptyset$ , then  $r'_{i_1 \dots i_k} = r_{i_1 \dots i_k}$ , or
  - if  $r'_{i_1 \dots i_k} \in \delta_{i_1 \dots i_k}(r_{i_1 \dots i_k})$ , but none of the rules  $\{r_{i_1}, \dots, r_{i_k}\}$  corresponding to  $r_{i_1 \dots i_k}$  are applicable to  $u_h$ , then  $u'_h = u_h$ .

If none of the cases above holds, that is, none of the rules corresponding to  $r_{i_1 \dots i_k}$  is applicable to  $u_h$ , and  $\delta_{i_1 \dots i_k}(r_{i_1 \dots i_k}) = \emptyset$ , then  $\delta_h(u_h, r_{i_1 \dots i_k}) = \emptyset$ , thus, the state  $(u_h, r_{i_1 \dots i_k})$  is a halting state in  $M_h$ .

Based on the above steps, we can construct a finite transition system  $M_h$  for each FIN-representable region  $h$ , such that the states of  $M_h$  correspond to the possible pairs of contents and collections of applicable rules, and the state sequences

of  $M_h$  correspond to the sequences of dynamically changing pairs of possible membrane contents and the corresponding rule collections.

If  $\Pi = (O, T, \mu, w_s, \langle w_{1L}, w_{1R} \rangle, \dots, \langle w_{nL}, w_{nR} \rangle, s)$  with  $1L, 1R, \dots, nL, nR, l \leq n$  being the labels of those regions which are directly enclosed in the skin membrane, and we let  $M_\Pi = M_{1 \dots l}$ , then the states of  $M_\Pi$  represent the dynamically changing collections of rules applicable in the skin region which can change as allowed by the possible transitions of  $M_\Pi$ . In short,  $M_\Pi$  represents the rule configurations of  $\Pi$ . □

**Example 4** Let us construct the representations of finitely representable membranes  $M_{\Pi_3} = M_1$  for the P system of Example 3. Starting with the elementary membranes, we get  $M_{2L} = \{(a, \emptyset), (a, \emptyset), \delta_{2L}\}$ ,  $M_{2R} = \{(c, \emptyset), (c, \emptyset), \delta_{2R}\}$  such that  $\delta_{2L}(a, \emptyset) = \delta_{2R}(c, \emptyset) = \emptyset$ , and  $M_2 = (R_2, \bar{r}_2, \delta_2)$  with  $R_2 = \{(a, \emptyset)\} \times \{(c, \emptyset)\} = \{(a, \emptyset), (c, \emptyset)\} = \{\bar{r}_2\}$ , and  $\delta_2(\bar{r}_2) = \emptyset$ . Similarly, we can construct  $M_i = (R_i, \bar{r}_i, \delta_i)$  for all  $i, 2 < i \leq 8$ , which all have a similar structure.

Now, given the transition systems  $M_2, \dots, M_5$  we construct  $M_{1L}$  as follows. We start with the construction of  $M_{2 \dots 5} = (R_{2 \dots 5}, \bar{r}_{2 \dots 5}, \delta_{2 \dots 5})$  as

- $R_{2 \dots 5} = \{\bar{r}_{2 \dots 5}\}$  where
- $\bar{r}_{2 \dots 5} = (\bar{r}_2, \dots, \bar{r}_5)$  with  $\bar{r}_2 = ((a, \emptyset), (c, \emptyset))$ ,  $\bar{r}_3 = ((a, \emptyset), (b, \emptyset))$ ,  $\bar{r}_4 = ((b, \emptyset), (a, \emptyset))$ ,  $\bar{r}_5 = ((c, \emptyset), (d, \emptyset))$ , and
- $\delta_{2 \dots 5}(\bar{r}_{2 \dots 5}) = \emptyset$ .

Note that the rule set corresponding to  $\bar{r}_{2 \dots 5}$  (using the rule labels from Example 3 to denote the rules) is  $\{r_2, r_3, r_4, r_5\} = \{a \rightarrow c, a \rightarrow b, b \rightarrow a, c \rightarrow d\}$ .

Now we can construct  $M_{1L} = (Q_{1L}, \bar{q}_{1L}, \delta_{1L})$  as follows. The set of possible states is  $Q_{1L} = \{a, b, c, d\} \times \{(\bar{r}_2, \bar{r}_3, \bar{r}_4, \bar{r}_5)\}$ , that is,

$$Q_{1L} = \{(a, (\bar{r}_2, \bar{r}_3, \bar{r}_4, \bar{r}_5)), (b, (\bar{r}_2, \bar{r}_3, \bar{r}_4, \bar{r}_5)), (c, (\bar{r}_2, \bar{r}_3, \bar{r}_4, \bar{r}_5)), (d, (\bar{r}_2, \bar{r}_3, \bar{r}_4, \bar{r}_5))\},$$

the initial state is  $\bar{q}_{1L} = (a, (\bar{r}_2, \bar{r}_3, \bar{r}_4, \bar{r}_5))$ , and the transition mapping is defined as

$$\begin{aligned} \delta_{1L}(a, (\bar{r}_2, \bar{r}_3, \bar{r}_4, \bar{r}_5)) &= \{(c, (\bar{r}_2, \bar{r}_3, \bar{r}_4, \bar{r}_5)), (b, (\bar{r}_2, \bar{r}_3, \bar{r}_4, \bar{r}_5))\}, \\ \delta_{1L}(c, (\bar{r}_2, \bar{r}_3, \bar{r}_4, \bar{r}_5)) &= \{(d, (\bar{r}_2, \bar{r}_3, \bar{r}_4, \bar{r}_5))\}, \\ \delta_{1L}(b, (\bar{r}_2, \bar{r}_3, \bar{r}_4, \bar{r}_5)) &= \{(a, (\bar{r}_2, \bar{r}_3, \bar{r}_4, \bar{r}_5))\}, \\ \delta_{1L}(d, (\bar{r}_2, \bar{r}_3, \bar{r}_4, \bar{r}_5)) &= \emptyset. \end{aligned}$$

With a similar construction, we can construct  $M_{1R} = (Q_{1R}, \bar{q}_{1R}, \delta_{1R})$  as

$$Q_{1R} = \{(aa, (\bar{r}_6, \bar{r}_7, \bar{r}_8)), (bb, (\bar{r}_6, \bar{r}_7, \bar{r}_8)), (be, (\bar{r}_6, \bar{r}_7, \bar{r}_8)), (ae, (\bar{r}_6, \bar{r}_7, \bar{r}_8)), (ee, (\bar{r}_6, \bar{r}_7, \bar{r}_8))\},$$

where we use a similar notation as above with  $\bar{r}_6 = ((a, \emptyset), (b, \emptyset))$ ,  $\bar{r}_7 = ((b, \emptyset), (a, \emptyset))$ ,  $\bar{r}_8 = ((a, \emptyset), (e, \emptyset))$ , and

$$\begin{aligned}\delta_{1R}(aa, (\bar{r}_6, \bar{r}_7, \bar{r}_8)) &= \{(bb, (\bar{r}_6, \bar{r}_7, \bar{r}_8)), \\ &\quad (be, (\bar{r}_6, \bar{r}_7, \bar{r}_8)), (ee, (\bar{r}_6, \bar{r}_7, \bar{r}_8))\}, \\ \delta_{1R}(bb, (\bar{r}_6, \bar{r}_7, \bar{r}_8)) &= \{(aa, (\bar{r}_6, \bar{r}_7, \bar{r}_8))\}, \\ \delta_{1R}(be, (\bar{r}_6, \bar{r}_7, \bar{r}_8)) &= \{(ae, (\bar{r}_6, \bar{r}_7, \bar{r}_8))\}, \\ \delta_{1R}(ae, (\bar{r}_6, \bar{r}_7, \bar{r}_8)) &= \{(be, (\bar{r}_6, \bar{r}_7, \bar{r}_8)), (ee, (\bar{r}_6, \bar{r}_7, \bar{r}_8))\}, \\ \delta_{1R}(ee, (\bar{r}_6, \bar{r}_7, \bar{r}_8)) &= \emptyset.\end{aligned}$$

Now, given  $M_{1L}$  and  $M_{1R}$  we can construct  $M_1 = M_{\Pi_3}$  as  $M_1 = (R_1, \bar{r}_1, \delta_1)$  where

$$R_1 = Q_{1L} \times Q_{1R},$$

is the set of states, and  $\bar{r}_1 = ((a, (\bar{r}_2, \bar{r}_3, \bar{r}_4, \bar{r}_5)), (aa, (\bar{r}_6, \bar{r}_7, \bar{r}_8)))$  is the initial state corresponding to the rule  $a \rightarrow aa$ .

If we denote  $(\bar{r}_2, \bar{r}_3, \bar{r}_4, \bar{r}_5)$  and  $(\bar{r}_6, \bar{r}_7, \bar{r}_8)$  as  $\bar{r}_{2\dots 5}$  and  $\bar{r}_{6\dots 8}$ , respectively, then the transition relation is defined as follows.

$$\begin{aligned}\delta_1((a, \bar{r}_{2\dots 5}), (aa, (\bar{r}_{6\dots 8}))) &= \{(b, \bar{r}_{2\dots 5}), (bb, (\bar{r}_{6\dots 8})), \\ &\quad (b, \bar{r}_{2\dots 5}), (be, (\bar{r}_{6\dots 8})), \\ &\quad ((b, \bar{r}_{2\dots 5}), (ee, (\bar{r}_{6\dots 8}))), ((c, \bar{r}_{2\dots 5}), (bb, (\bar{r}_{6\dots 8}))), \\ &\quad ((c, \bar{r}_{2\dots 5}), (be, (\bar{r}_{6\dots 8}))), ((c, \bar{r}_{2\dots 5}), (ee, (\bar{r}_{6\dots 8})))\}, \\ \delta_1((b, \bar{r}_{2\dots 5}), (bb, (\bar{r}_{6\dots 8}))) &= \{((a, \bar{r}_{2\dots 5}), (aa, (\bar{r}_{6\dots 8})))\}, \\ \delta_1((b, \bar{r}_{2\dots 5}), (be, (\bar{r}_{6\dots 8}))) &= \{((a, \bar{r}_{2\dots 5}), (ae, (\bar{r}_{6\dots 8})))\}, \\ \delta_1((b, \bar{r}_{2\dots 5}), (ee, (\bar{r}_{6\dots 8}))) &= \{((a, \bar{r}_{2\dots 5}), (ee, (\bar{r}_{6\dots 8})))\}, \\ \delta_1((c, \bar{r}_{2\dots 5}), (bb, (\bar{r}_{6\dots 8}))) &= \{((d, \bar{r}_{2\dots 5}), (aa, (\bar{r}_{6\dots 8})))\}, \\ \delta_1((c, \bar{r}_{2\dots 5}), (be, (\bar{r}_{6\dots 8}))) &= \{((d, \bar{r}_{2\dots 5}), (ae, (\bar{r}_{6\dots 8})))\}, \\ \delta_1((c, \bar{r}_{2\dots 5}), (ee, (\bar{r}_{6\dots 8}))) &= \{((d, \bar{r}_{2\dots 5}), (ee, (\bar{r}_{6\dots 8})))\}, \\ \delta_1((a, \bar{r}_{2\dots 5}), (ae, (\bar{r}_{6\dots 8}))) &= \{((c, \bar{r}_{2\dots 5}), (be, (\bar{r}_{6\dots 8}))), \\ &\quad ((c, \bar{r}_{2\dots 5}), (ee, (\bar{r}_{6\dots 8}))), \\ &\quad ((b, \bar{r}_{2\dots 5}), (be, (\bar{r}_{6\dots 8}))), ((b, \bar{r}_{2\dots 5}), (ee, (\bar{r}_{6\dots 8})))\}, \\ \delta_1((a, \bar{r}_{2\dots 5}), (ee, (\bar{r}_{6\dots 8}))) &= \{((c, \bar{r}_{2\dots 5}), (ee, (\bar{r}_{6\dots 8}))), \\ &\quad ((b, \bar{r}_{2\dots 5}), (ee, (\bar{r}_{6\dots 8})))\}, \\ \delta_1((d, \bar{r}_{2\dots 5}), (aa, (\bar{r}_{6\dots 8}))) &= \{((d, \bar{r}_{2\dots 5}), (ee, (\bar{r}_{6\dots 8}))), \\ &\quad ((d, \bar{r}_{2\dots 5}), (be, (\bar{r}_{6\dots 8}))), \\ &\quad ((d, \bar{r}_{2\dots 5}), (bb, (\bar{r}_{6\dots 8})))\}, \\ \delta_1((d, \bar{r}_{2\dots 5}), (ae, (\bar{r}_{6\dots 8}))) &= \{((d, \bar{r}_{2\dots 5}), (ee, (\bar{r}_{6\dots 8}))), \\ &\quad ((d, \bar{r}_{2\dots 5}), (be, (\bar{r}_{6\dots 8})))\}, \\ \delta_1((d, \bar{r}_{2\dots 5}), (bb, (\bar{r}_{6\dots 8}))) &= \{((d, \bar{r}_{2\dots 5}), (aa, (\bar{r}_{6\dots 8})))\}, \\ \delta_1((d, \bar{r}_{2\dots 5}), (be, (\bar{r}_{6\dots 8}))) &= \{((d, \bar{r}_{2\dots 5}), (ae, (\bar{r}_{6\dots 8})))\}, \\ \delta_1((d, \bar{r}_{2\dots 5}), (ee, (\bar{r}_{6\dots 8}))) &= \emptyset.\end{aligned}$$

We can now prove the following theorem.

**Theorem 3**  $\mathcal{L}(OP(polym, ncoo, fin)) \subseteq PsETOL$ .

**Proof** Let  $\Pi = (O, T, \mu, w_s, \langle w_{1L}, w_{1R} \rangle, \dots, \langle w_{nL}, w_{nR} \rangle, s)$  be a polymorphic P system,  $\Pi \in OP(polym, ncoo, fin)$ , and let us assume (without loss of generality) that the membranes that are directly contained in the skin region are labeled by the labels  $1L, 1R, \dots, kL, kR, k \leq n$ . Since both the left- and right-hand membranes  $iL, iR, 1 \leq i \leq k$ , are FIN-representable, we can construct the transition system  $M_\Pi = (R_\Pi, \bar{r}_\Pi, \delta_\Pi) = M_{1\dots k} = (R_{1\dots k}, \bar{r}_{1\dots k}, \delta_{1\dots k})$  as described in the proof of Lemma 2. For any  $q = (r_1, \dots, r_k) \in R_\Pi$ , let us denote the set of rules corresponding to  $(r_1, \dots, r_k)$  by  $rules(q)$ .

Now, based on  $M_\Pi$ , we construct an ETOL system  $G = (V, T, U, w)$ , where  $V$  is the alphabet,  $T$  is the terminal alphabet with  $T \subseteq V$ ,  $w$  is the initial string, and  $U$  is a set of tables,  $U = \{P_q \mid q \in R_\Pi\} \cup \{P_q, P_{q,halt} \mid q \in R_\Pi \text{ is a halting state}\} \cup \{P_{halt}\}$  containing at most three tables for each state of  $M_\Pi$  and one additional table.

The ETOL system  $G$  is defined as follows:

$$V = \{a' \mid a \in O\} \cup \{D_q, D_q, D_{q,halt} \mid q \in R_\Pi\} \cup \{F, D_{halt}\} \cup T.$$

Given a state  $q \in R_\Pi$ , let us denote for a rule  $r : u \rightarrow v \in rules(q)$  by  $r' : u' \rightarrow v'$  the rule  $h(u) \rightarrow h(v)$  where if  $w = a_1 \dots a_t$ ,  $a_i \in O, 1 \leq i \leq t$ , then  $h(w) = h(a_1) \dots h(a_t)$  for  $h(a) = a' \in V$ .

The axiom is given as

$$w = D_{\bar{r}_\Pi} w'_s,$$

where  $w'_s = h(w_s)$ , the primed version of a string corresponding the initial contents of the skin region of  $\Pi$ .

We construct a table  $P_q$  for each state  $q \in R_\Pi$  as

$$\begin{aligned}P_q &= \{r' \mid r \in rules(q)\} \cup \{D_q \rightarrow D_s \mid s \in \delta_\Pi(q)\} \cup \\ &\quad \{D_q \rightarrow D_q \mid \delta_\Pi(q) = \emptyset\} \cup \\ &\quad \{D_z \rightarrow F \mid D_z \neq D_q\}.\end{aligned}$$

Note that the rules  $D_q \rightarrow D_q$  can be applied when  $q$  is a halting state. The rule  $D_z \rightarrow F$  ensures that the table is only applied when the computation is in the appropriate state, since the symbol  $F$  functions as a trap: if the rule  $D_z \rightarrow F$  is applied during the computation, the system will never produce a terminal string.

Note also that for each object  $x$  that does not have a rule above, we assume that  $x \rightarrow x$  is present in each table.

It must be checked that the table belonging to the halting state can only be used in the appropriate step. In order to achieve this, it is necessary to introduce new tables for each  $q \in R_\Pi$ , which can only be used if  $q$  is a halting state. For each halting state  $q$ , we construct the tables  $P_q$  and  $P_{q,halt}$  as follows. Let

$$P_{\underline{q}} = \{r' \mid r \in \text{rules}(q)\} \cup \{D_{\underline{q}} \rightarrow D_{\underline{q}}, D_{\underline{q}} \rightarrow D_{\underline{q},\text{halt}}\} \cup \{D_x \rightarrow F \mid D_x \neq D_{\underline{q}}\}.$$

Since  $q \in R_{\Pi}$  is a halting state, the rule configuration represented by  $q$  does not change in the P system  $\Pi$  any more during the computation. Thus,  $\Pi$  reaches a halting state, if and only if the rules represented by  $\underline{q}$  are not applicable in the skin membrane.  $G$  guesses that this is the case by introducing the symbol  $D_{\underline{q},\text{halt}}$ . After applying the rule  $D_{\underline{q}} \rightarrow D_{\underline{q},\text{halt}}$ , the table  $P_{\underline{q},\text{halt}}$  makes sure that the current rules are not applicable by introducing trap symbols if any of them are. This is achieved by

$$P_{\underline{q},\text{halt}} = \{u' \rightarrow F \mid u \rightarrow v \in \text{rules}(q)\} \cup \{D_{\underline{q},\text{halt}} \rightarrow D_{\text{halt}}\} \cup \{D_x \rightarrow F \mid D_x \neq D_{\underline{q},\text{halt}}\}.$$

In the case of the corresponding configuration of  $\Pi$  being a non-halting configuration, the use of this table causes all rewritable symbols to be rewritten to  $F$  and the computation will not produce any result.

In the case of a proper halting configuration (if the symbol  $F$  is not introduced), an additional table, table  $P_{\text{halt}}$  must be used. This table deletes the marker symbols and rewrites the remaining letters to terminals as follows.

$$P_{\text{halt}} = \{X \rightarrow \lambda \mid X \in O \setminus \{a' \mid a \in T\}\} \cup \{a' \rightarrow a \mid a \in T\} \cup \{D_x \rightarrow F \mid D_x \neq D_{\text{halt}}\}.$$

By the construction above,  $a_1 a_2 \dots a_k \in L(\Pi)$  implies that  $a_1 a_2 \dots a_k$  is an element of  $L(G)$ . The other way around, we may also see that if  $a_1 a_2 \dots a_k$  is successfully produced by  $G$ , then  $a_1 a_2 \dots a_k$  must also be an element of  $L(\Pi)$ .  $\square$

**Corollary 4**  $\mathcal{L}(OP(\text{polym}, \text{ncoo}, \text{fin})) = PsETOL$ .

**Proof** By observing the proof of Theorem 1, we may see that for any ETOL system  $G$ , the right-hand regions of the P system  $\Pi \in OP^3(\text{polym}, \text{ncoo})$  constructed to simulate  $G$  are FIN-representable. Combining this observation with Theorem 3 we obtain our statement.  $\square$

**Example 5** Consider the polymorphic P system of Example 3 which can be seen in Fig. 3. Let us construct an ETOL system  $G = (V, T, U, w)$  simulating this membrane system based on the transition system  $M_{\Pi_3} = M_1 = (R_1, \bar{r}_1, \delta_1)$  from

Example 4. The nonterminal alphabet of  $G$  includes  $O$ , a set of labeled symbols and trap symbol in addition, the terminals correspond to the terminal objects of the P system, and the axiom corresponds to the initial contents of the skin region and the initial rule configuration of the P system.

Recall that according to the above construction, we need the nonterminals  $D_q, D_{\bar{q}}$ , and  $D_{\bar{q},\text{halt}}$  in the alphabet of the ETOL system for all states  $q \in R_1$ , and for the initial state, for example, these would be written as  $D_{((a,\bar{r}_2\dots_5),(aa,\bar{r}_6\dots_8))}$ ,  $D_{((a,\bar{r}_2\dots_5),(aa,\bar{r}_6\dots_8))}$ , and  $D_{((a,\bar{r}_2\dots_5),(aa,\bar{r}_6\dots_8)),\text{halt}}$ . Since the rules corresponding to the (elementary) membranes  $2L, 2R, \dots, 8L, 8R$ , that is,  $\bar{r}_{2\dots 5} = (\bar{r}_2, \dots, \bar{r}_5)$  and  $\bar{r}_{6\dots 8} = (\bar{r}_6, \dots, \bar{r}_8)$  do not change during the computation, for the sake of simplicity, we use the symbols  $D_{a \rightarrow aa}, D_{a \rightarrow aa}$ , and  $D_{a \rightarrow aa,\text{halt}}$ , that is,  $D_r, D_r$ , and  $D_{r,\text{halt}}$  for  $r \in \text{rules}(q)$  instead of the more general and precise notation that would be necessary for more complicated membrane structures. Thus, we have

$$V = \{a, a', b, b', c, c', d, d', e, e'\} \cup \{D_r, D_r, D_{r,\text{halt}} \mid r \in \text{rules}(q), q \in R_1\} \cup \{D_{\text{halt}}, F\},$$

$$T = \{a, b, c, d, e\},$$

$$w = D_{a \rightarrow aa} a'.$$

We start by constructing the first table which contains the current instance of the rule associated to the skin membrane, that is, the rule corresponding to  $\bar{r}_1$ , the relabeling rules, and additional rules for identical rewriting of the other symbols (because ETOL systems rewrite every symbol at every step) which we do not indicate in the tables below, for the sake of brevity.

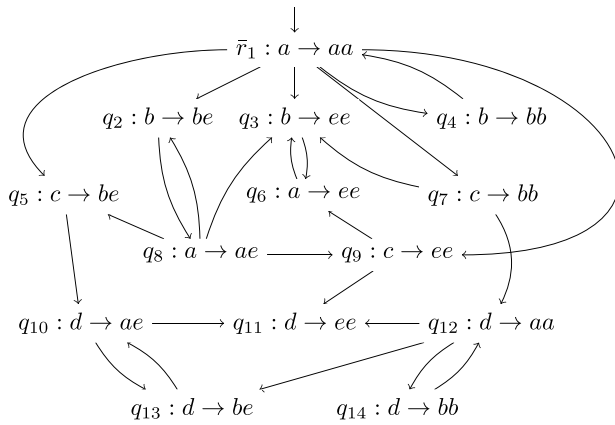
The construction of the relabeling rules is based on the number of different configurations that can be obtained after the next computational step. Since there are six possible configurations that the P system can reach in the first step, the symbol  $D_{a \rightarrow aa}$  can be relabeled in six ways, either to  $D_{b \rightarrow bb}, D_{c \rightarrow bb}, D_{b \rightarrow be}, D_{c \rightarrow be}, D_{b \rightarrow ee},$  or  $D_{c \rightarrow ee}$ .

$$P_{\bar{r}_1} = \{D_{a \rightarrow aa} \rightarrow D_{\alpha \rightarrow \beta} \mid \alpha \in \{b, c\}, \beta \in \{bb, be, ee\}\} \cup \{D_z \rightarrow F \mid D_z \neq D_{a \rightarrow aa}\} \cup \{a' \rightarrow a' a'\}.$$

Based on similar considerations, we construct tables for each non-halting state  $q \in R_1$ .

$$P_q = \{D_r \rightarrow D_{r'}, D_z \rightarrow F \mid r \in \text{rules}(q), r' \in \text{rules}(q') \text{ for } q' \in \delta_1(q), z \neq r\} \cup \{\alpha' \rightarrow \beta' \mid \alpha \rightarrow \beta \in \text{rules}(q)\}.$$

As  $q_{11}$  with  $d \rightarrow ee \in \text{rules}(q_{11})$  is a halting state of  $M_1$ , we have



**Fig. 5** The graph structure representing the order in which the tables constructed in Example 4 can be applied after each other. The labels denote the states of  $M_{11}$  and the corresponding rules

$$P_{d \rightarrow ee} = \{D_{d \rightarrow ee} \rightarrow D_{d \rightarrow ee}, D_z \rightarrow F \mid z \neq d \rightarrow ee\} \cup \{d' \rightarrow e'e'\},$$

$$P_{d \rightarrow ee} = \{D_{d \rightarrow ee} \rightarrow D_{d \rightarrow ee}, D_{d \rightarrow ee} \rightarrow D_{d \rightarrow ee, halt}, D_z \rightarrow F \mid z \neq d \rightarrow ee\} \cup \{d' \rightarrow e'e'\},$$

$$P_{d \rightarrow ee, halt} = \{D_{d \rightarrow ee, halt} \rightarrow D_{halt}, D_z \rightarrow F \mid z \neq d \rightarrow ee, halt\} \cup \{d' \rightarrow F\},$$

and finally

$$P_{halt} = \{D_{halt} \rightarrow \lambda, D_z \rightarrow F \mid z \neq halt\} \cup \{x' \rightarrow x \mid x \in \{a, b, c, d, e\}\}.$$

Figure 5 shows a graph which is based on the order in which the application of the tables constructed above can be applied.

### 5 Conclusion

We have shown how ETOL systems can be simulated by restricted variants of non-cooperative polymorphic P system of depth three, then showed that the simulation also works the other way around, even if the depth of the simulated P systems is not limited, but the regions are finitely representable. Thus, a precise characterization of Parikh sets of ETOL languages can be obtained in term of polymorphic P systems. Our work is intended to be an initial step in the investigation of the computing power of non-cooperative polymorphic systems with limited depth or FIN-representable regions.

**Funding** Open access funding provided by University of Debrecen.

**Data availability** No datasets were generated or analysed during the current study.

### Declarations

**Conflict of interest** The authors have no relevant financial or non-financial interests to disclose.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

### References

- Păun, G. (2000). Computing with membranes. *Journal of Computer and System Sciences*, 61(1), 108–143. <https://doi.org/10.1006/jcss.1999.1693>.
- Păun, G. (2002). *Membrane computing: An introduction*. Berlin, Heidelberg: Springer. <https://doi.org/10.1007/978-3-642-56196-2>.
- Păun, G., Rozenberg, G., & Salomaa, A. (Eds.). (2010). *The Oxford handbook of membrane computing*. Oxford: Oxford University Press.
- Freund, R. (1999). Generalized P-systems. In G. Ciobanu & G. Păun (Eds.), *Fundamentals of computation theory* (Lecture Notes in Computer Science, vol. 1684, pp. 281–292). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/3-540-48321-7\\_23](https://doi.org/10.1007/3-540-48321-7_23).
- Arroyo, F., Baranda, A. V., Castellanos, J., & Păun, G. (2002). Membrane computing: the power of (rule) creation. *Journal of Universal Computer Science*, 8(3), 369–381. <https://doi.org/10.3217/jucs-008-03-0369>.
- Cavaliere, M., Ionescu, M., & Ishdorj, T.-O. (2005). Inhibiting/de-inhibiting rules in P systems. In G. Mauri, G. Păun, M. J. Pérez-Jiménez, G. Rozenberg, & A. Salomaa (Eds.), *Membrane computing* (Lecture Notes in Computer Science, vol. 3365, pp. 224–238). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/978-3-540-31837-8\\_13](https://doi.org/10.1007/978-3-540-31837-8_13).
- Alhazov, A., Ivanov, S., & Rogozhin, Y. (2011). Polymorphic P systems. In M. Gheorghe, T. Hinze, G. Păun, G. Rozenberg, & A. Salomaa (Eds.), *Membrane computing* (Lecture Notes in Computer Science, vol. 6501, pp. 81–94). Berlin, Heidelberg: Springer.
- Alhazov, A., Ivanov, S., & Freund, R. (2016). Polymorphic P systems: A survey. *Bulletin of the International Membrane Computing Society*, 2, 79–101.
- Ivanov, S. (2014). Polymorphic P systems with non-cooperative rules and no ingredients. In M. Gheorghe, G. Rozenberg, A. Salomaa, P. Sosík, & C. Zandron (Eds.), *Membrane computing* (Lecture notes in computer science, vol. 8961, pp. 258–273). Cham: Springer.
- Román, G. (2019). Inference of bounded L systems with polymorphic P systems. *Journal of Membrane Computing*, 1, 52–57. <https://doi.org/10.1007/s41965-019-00007-0>.

11. Rozenberg, G., & Salomaa, A. (Eds.). (1997). *Handbook of formal languages*. Berlin Heidelberg: Springer.
12. Lindenmayer, A. (1968). Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. *Journal of Theoretical Biology*, 18(3), 280–299. [https://doi.org/10.1016/0022-5193\(68\)90079-9](https://doi.org/10.1016/0022-5193(68)90079-9).
13. Rozenberg, G., & Salomaa, A. (1992). *Lindenmayer systems: Impacts on theoretical computer science, computer graphics, and developmental biology*. Berlin, Heidelberg: Springer.
14. Ehrenfeucht, A., Rozenberg, G., & Skyum, S. (1976). A relationship between ETOL and EDTOL languages. *Theoretical Computer Science*, 1(4), 325–330. [https://doi.org/10.1016/0304-3975\(76\)90076-1](https://doi.org/10.1016/0304-3975(76)90076-1).

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**György Vaszil** obtained his PhD in 2001 at the Eötvös Loránd University of Budapest. Since 2015, he is full professor at the Faculty of Informatics of the University of Debrecen where he is the head of the Department of Computer Science. His research interests include the theory of formal languages and automata, unconventional or nature motivated computational models, such as bio-inspired models like membrane systems. He has published more than 150 papers in international journals, confer-

ences, and workshops.



**Anna Kuczik** graduated in computer science from the Faculty of Informatics of the University of Debrecen in 2022. She started her PhD in the same year at the Doctoral School of Informatics. Her current research interests include unconventional computational models and their applications, membrane computing and rough set theory.