Natural Computing (2023) 22:13–25
https://doi.org/10.1007/s11047-022-09925-y

On the power of boundary rule application in membrane computing

Péter Battyányi[1]

Accepted: 24 October 2022 / Published online: 9 November 2022

Abstract
In this paper, it is investigated how different features of membrane systems can be simulated by the boundary rule application. Firstly, it is discussed how the effect of maximally parallel mode can be obtained by non-cooperative boundary rules applied only in sequential mode, then it is also demonstrated how membrane dissolution, or the application of using promoters and inhibitors can be simulated.

Keywords Symbol object P system · Multiset approximation space · Rough set · Maximal parallel mode · Non-cooperative rule

1 Introduction

Membrane systems, introduced in Păun (2000), are biologically inspired models of computation: similarly to the cells functioning in an organism, the membranes exhibit independent behaviour and, on the contrary, the whole system shows a synchronized computational manner. The computation proceeds in distinct regions, which are called membranes or compartments. The membranes contain multisets and each computational step means the local evolution of multisets in the membranes obeying certain rules. In each big step of computation we assume that the membranes act in a maximally parallel manner, that is, as many multiset rules are performed simultaneously in each membrane as possible starting from the actual membrane content. The computation is synchronized: at the end of a big operational step the membranes wait for the others to finish their maximal computational step and the next cycle begins only after that. Concerning the computational mode, several variants of P systems have been introduced and studied, see the monograph (Păun 2002) for a thorough introduction, or the handbook (Păun et al. 2010) for a summary of notions and results of the area.

✉ Péter Battyányi
battyanyi.peter@inf.unideb.hu

[1] Department of Computer Science, Faculty of Informatics, University of Debrecen, Kassai út 26, Debrecen 4028, Hungary

The question of how to define dynamically changing membrane structures using topological spaces, and how the underlying topologies influence the behaviour of P systems was already examined in Csuhaj-Varjú et al. (2012, 2015). Multiset approximation spaces were defined in Mihálydeák and Csajbók (2013, 2014), which made it possible to talk about lower and upper approximations of the contents of the membranes of a P system. This also opened the door to various definitions of the membrane rules: permitting only rules that allow the manipulation of multisets being on the boundaries of the membranes or working only with rules in the inner parts of the membranes led to membrane systems of different computational strengths (Battyányi and Vaszil 2019). The first results in this area appeared in Mihálydeák and Vaszil (2015), where symport/antiport rules were examined together with a notion of closeness of membranes. The investigations were continued in Battyányi et al. (2019), where it was proved that, for generalized P systems, certain topological restrictions on rule applications are itself enough to ensure Turing completeness, that is, we are able to substitute maximal parallelism with relying only on rules defined on the boundaries provided the respective membranes are close to each other in some sense. In Battyányi and Vaszil (2019), we also examined generalized P systems, but we managed to free ourselves from the notion of closeness of membranes and we leaned merely on the various forms of rule applications. We considered a rule executable only if the left hand side is either in the inner part or on the boundary of the compartment, respectively. It turned out that it is only the use

of the boundary rules, i. e., the rules which can only manipulate objects on the boundaries of the compartments, that resulted in a computational model equivalent in strength to Turing machines.

In this paper, it is examined how the idea of boundary rule application can be used to obtain the effects of membrane system features like maximally parallel rule application, membrane dissolution, or the use of promoters/inhibitors in the usual symbol object membrane systems. It is proven that boundary rules are in themselves enough to compensate for maximally parallel rule application in symbol object membrane systems, and, moreover, membrane dissolution and the applications of promoters/inhibitors can also be simulated by choosing an appropriate approximation space. In addition, all the membrane systems presented here operate with non-cooperative rules in sequential mode, which is an improvement compared to the previous results.

2 Preliminaries

Let $\mathbb{N}$ and $\mathbb{N}_{>0}$ demote the set of non-negative integers and the set of positive integers, respectively, and let O be a finite nonempty set, which is called the set of objects. A *multiset* M over O is a pair $M = (O,f)$, where the mapping $f : O \to \mathbb{N}$ gives the *multiplicity* of each object $a \in O$. The set $\mathrm{supp}(M) = \{a \in O \mid f(a) > 0\}$ is called the *support* of M. If $\mathrm{supp}(M) = \emptyset$, then M is the empty multiset. If $a \in \mathrm{supp}(M)$, then $a \in M$, and $a \in^n M$ if $f(a) = n$. If I is an index set and a_i are objects $(i \in I)$, then we denote the set formed by the elements a_i by $\{a_i \mid i \in I\}$, as usual. The number of elements in a multiset M is denoted by $|M|$.

Let $M_1 = (O,f_1), M_2 = (O,f_2)$. Then $(M_1 \sqcap M_2) = (O,f)$ where $f(a) = \min\{f_1(a),f_2(a)\}$; $(M_1 \sqcup M_2) = (O,f')$, where $f'(a) = \max\{f_1(a),f_2(a)\}$; $(M_1 \oplus M_2) = (O,f'')$, where $f''(a) = f_1(a) + f_2(a)$; $(M_1 \ominus M_2) = (O,f''')$ where $f'''(a) = \max\{f_1(a) - f_2(a), 0\}$; and $M_1 \sqsubseteq M_2$, if $f_1(a) \leq f_2(a)$ for all $a \in O$.

For any $n \in \mathbb{N}$, n-times addition of M, denoted by $\oplus_n M$, is given by the following inductive definition:

- $\oplus_0 M = \emptyset$;
- $\oplus_1 M = M$;
- $\oplus_{n+1} M = (\oplus_n M) \oplus M$.

Let $M_1 \neq \emptyset, M_2$ be two multisets. For any $n \in \mathbb{N}$, $M_1 \sqsubseteq^n M_2$, if $\oplus_n M_1 \sqsubseteq M_2$ but $\oplus_{n+1} M_1 \not\sqsubseteq M_2$. In case of $M_1 = \emptyset$, let $M_1 \sqsubseteq^n M_2$ be false for any multiset M_2 and any $n \in \mathbb{N}$.

The number of copies of objects in a finite multiset $M = (O,f)$ is its cardinality: $\mathrm{card}(M) = \Sigma_{a \in \mathrm{supp}(M)} f(a)$. Such an M can be represented by any string w over O for which $|w| = \mathrm{card}(M)$, and $|w|_a = f(a)$ where $|w|$ denotes the length of the string w, and $|w|_a$ denotes the number of occurrences of symbol a in w.

We define the $\mathcal{MS}^n(O)$, $n \in \mathbb{N}$, to be the set of all multisets $M = (O,f)$ over O such that $f(a) \leq n$ for all $a \in O$, and we let $\mathcal{MS}^{<\infty}(O) = \bigcup_{n \geq 0} \mathcal{MS}^n(O)$. $\mathcal{MS}^{<\infty}(O)$ is abbreviated by $\mathcal{MS}(O)$ unless stated otherwise.

2.1 Multiset approximation spaces

In his seminal papers (Pawlak 1982, 1991), Pawlak proposed rough set theory as a theory of dealing with imprecise, partial information. The basic concept relied on an equivalence relation defined on the objects of the underlying set, the objects belonging to the same partition with respect to that equivalence relation were considered to be indiscernable. This means, one could observe the equivalence class as a whole without being able to distinguish its elements. Any union of these partitions constituted a definable set. The question was how it was possible to describe any subset of the universe by the known information, i.e., by definable sets. A lower approximation could be considered as the union of all definable sets contained in the underlying set, while the upper approximation is the union of all definable sets which have nonempty intersection with the set in question. Deriving from Pawlak's original notion of rough sets various ways of set approximations in rough set theory have been proposed. One possibility is to define approximations based on a set of base sets and, without leaning on an underlying equivalence relation, one can construct the definable sets starting from the base sets. The lower and upper approximations can be constructed for any set in a way similar to the original method. This natural generalization of set approximation was defined in Mihálydeák and Csajbók (2013). The lower and upper approximations rely on base sets which can be thought of as representants of the available knowledge: we are able to discover the elements of each set only to the amount of its constituent or bordering base sets. In possession of the concepts of lower and upper approximations, we can also introduce the concept of boundary as the difference between these two.

The set approximation framework for multisets is generalized in order to be able to define the respective approximation multisets for membranes. The notion of multiset approximation spaces has been introduced in Mihálydeák and Csajbók (2013) (see also Mihálydeák and Csajbók 2014 for more details). We adopt the definitions there to our present situation.

A *multiset approximation space* is a quintuple $(O, \mathfrak{B}, \mathsf{l}, \mathsf{u}, \mathsf{b})$ where O is a finite set, $\mathfrak{B} \subseteq \mathcal{MS}^{<\infty}(O)$ is a base system (a set of base multisets). We assume none of

the base sets are empty. The functions $\mathsf{b}, \mathsf{u}, \mathsf{l} : \mathcal{MS}^{<\infty}(O) \to \mathcal{MS}^{<\infty}(O)$ are termed the approximation functions generated by $\mathfrak{B}$.

For any multiset $M = (O,f) \in \mathcal{MS}^{<\infty}(O)$, the *lower approximation function* can be defined as:

$$\mathsf{l}(M) = \bigsqcup\{\oplus_n B \mid B \in \mathfrak{B} \text{ and } B \sqsubseteq^n M\}.$$

The *boundary function* can be written as:

$$\mathsf{b}(M) = \bigsqcup\{\oplus_n B \mid B \in \mathfrak{B} \text{ and } B \sqcap (M \ominus \mathsf{l}(M)) \sqsubseteq^n M \ominus \mathsf{l}(M)\},$$

and and the *upper approximation function* is:

$$\mathsf{u}(M) = \mathsf{l}(M) \sqcup \mathsf{b}(M).$$

In addition, we also define the the *internal part* of the boundary of M as

$$\mathsf{b}^{\mathsf{i}}(M) = \mathsf{b}(M) \sqcap M.$$

Let $\mathfrak{N} \subseteq \mathcal{MS}^{<\infty}(O)$ be finite and assume $D \sqsubseteq \sqcup \mathfrak{N}$. Then D is *fully observable* with respect to $\mathfrak{N}$ if, for every $N \in \mathfrak{N}$, if $D \sqcap N \neq \emptyset$, then $D \sqcap N \sqsubseteq b^i(N)$. (The case $D \sqcap N = \emptyset$ is not relevant.) In other words, every submultiset of D that is contained in a multiset $N \in \mathfrak{N}$ is on the boundary of N. We call a nultiset u *observable* with respect to $\mathfrak{N}$ if there exists a multiset D fully observable with respect to $\mathfrak{N}$ and $N \in \mathfrak{N}$ such that $u \sqsubseteq D \sqcap N$. We say that u is on the *observable boundary* of N in this case or, on the observable boundary in short. We may omit the phrase "with respect to $\mathfrak{N}$" if it is clear from the context. Observe that every fully observable base set is the finite union of observable sets.

Let us consider an example illuminating the key concepts discussed here.

Example 1 Let $O = \{a,b,c,d,e\}$ and $M_1 = a^4b^2c$, $M_2 = d^2e$. Assume $\mathfrak{B} = \{ab, a^2c, de, ae, bd\}$. Then

$$l(M_1) = a^2b^2 \sqcup a^2c = a^2b^2c,$$

and

$$b(M_1) = a^2b^2 \sqcup a^2c \sqcup a^2e^2 = a^2b^2ce^2.$$

Similarly,

$$l(M_2) = de,$$
$$b(M_2) = bde.$$

Let $\mathfrak{N} = \{M_1, M_2\}$. Then, for example, $B_1 = a^2b^2de$ is fully observable with respect to $\mathfrak{N}$, while $B_2 = a^2b^2d^2e$ is not.

2.2 Regulating rule application in the multiset approximation framework

In Battyányi et al. (2019), P systems with dynamical structure were considered where the dynamic character of the membrane system was encoded in the reformulation of the region structure regarding a closeness property defined among the membranes based on the actual configuration of the system. In Battyányi and Vaszil (2019), we examined what kind of questions arise when we require that in order for a rule to be applicable in a generalized P system, the multisets on its left hand side must conform to certain properties defined in the multiset approximation framework associated with the system. We also considered the approach demanding that the multisets on the left hand sides of the rules should come from the boundaries of the respective compartments. We came to the conclusion that rules from the inner part ensure only limited computational capacity, while the rules on the boundaries lend full computational strength to the P system. That is, membrane systems with boundary rules are equivalent in computational strength to the general Turing machine concept. The present paper deals with symbol object membrane systems and even improves the previous results by demanding only non-cooperative rules to ensure the full computational power of the original P systems. We also give a direct construction of how to eliminate additional computational features like membrane dissolution and promoters/inhibitors for the P system by keeping their computational strength at the same time. The constructions we implement here are making use of non-cooperative rules in sequential execution mode or at least can be led back to computational models of that kind. This involves that the result of the simulations could yield a computational model that is in a different complexity class than the original P system, though the computed multisets are the same, since every parallel action of the original P system is modeled in our devices by a series of sequential computational steps.

3 The power of boundary rules

In this section, we prove that a symbol object membrane system with the maximally parallel mode can be simulated by a membrane system with the boundary rules in sequential mode. In addition, the rules of the newly constructed membrane system are non-cooperative rules. Below, we give the usual definition of P systems, though, in the subsequent sections, a little bit informal definition is sufficient, as we will see later.

Definition 1

1. A *P system* of degree $n \geq 1$ is $\Pi = (O, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n)$ where O is an alphabet of objects, μ is a membrane structure of n membranes, and let $par(m_j) = m_i$ denote that the parent membrane of m_j is m_i if $m_j \neq m_1$, $w_i \in \mathcal{MS}(O)$ with $1 \leq i \leq n$ are the

initial contents of the n regions, R_i with $(1 \leq i \leq n)$ are the sets of evolution rules associated with the regions; they are of the form $u \to v$, where $u \in \mathcal{MS}(O)$ and $v \in \mathcal{MS}^{<\infty}(O \times \{tar\})$, where $tar \in \{here, out\} \cup \{in_j \mid 1 \leq j \leq n\}$. We name membrane m_1 the *skin* membrane.

2. Let $r = u \to v \in R_i$ for some $i \in \{1, \ldots, n\}$. We collect the objects a in the multiset v_1 such that $(a, here) \in rhs(r)$. Similarly, we collect the objects b in the multiset v_2 such that $(b, out) \in rhs(r)$. Finally, we collect the objects c in the multiset v_3 such that $(c, in_j) \in rhs(r)$ for some $2 \leq j \leq n$ and $par(j) = i$. With this notation, we write $r = (v_1, here)(v_2, out)(v_3, in)$ for r, or, more briefly, $r = v_1(v_2, out)(v_3, in)$.

We discuss the semantics of the above membrane system. The P system changes its configuration by performing a computational step, which consists of the following actions. Firstly, a configuration is a tuple $C = (w_1, \ldots, w_n)$, where w_i are the multiset content of compartment m_i $(1 \leq i \leq n)$. A computational step is a transition from configuration C to configuration C', in notation $C \Rightarrow C'$, which is implemented in the following way. Let m_i be an arbitrary region, consider $r = u \to v \in R_i$. If $lhs(r) \sqsubseteq w_i$, we subtract the multiset u from w_i. The rule applications follow a maximally parallel mode, that is, we subtract the left hand side of a rule until no more multisets can be subtracted. We may use the same rule arbitrary many times. In the same time we collect the objects standing on the right hand sides of the rules. Having finished the subtraction of the left hand sides, we add the right hand sides to the respective regions in the following way. If we have an object (a, *here*), then we add a to membrane m_i. Furthermore, if we have an object (a, *out*), then we add a to the parent membrane of m_i provided m_i is not the skin membrane. In this case a is sent out into the environment. Finally, an object (a, in_j) means adding a to m_j, where $par(m_j) = m_i$. When all objects with target indication are moved to their correct place, then the new configuration C' is obtained and we are able to initiate the next computational step. For a more complete description, the reader is referred to Freund and Verlan (2007) and Păun (2002).

As was mentioned before, we assume that m_1 is the skin membrane and, additionally, no object in O evolves in the skin membrane. For the sake of simplicity and in order to demonstrate the difference that may sometimes occur between the treatment of elementary and non-elementary membranes in the subsequent constructions, we assume that m_n is an elementary membrane, that is, a membrane containing no submembranes. This assumption does not reduce the validity of our argumentation.

In the next section, additional features of the membrane system are considered. First of all, we may assume that a membrane can disappear in the course of the computation. Namely, the set of objects is extended with an additional object δ. If δ appears in a rule $r \in R_i$ $(1 \leq i \leq n)$, then the computational step is performed as above, but after that, as the effect of the presence of δ, the region m_i together with its set of rules disappear from the P system. This means that the objects of m_i are passed over to the parent region (except for δ, which disappears), and the rules in R_i are not applied anymore. Note that the outermost region (the skin region) cannot dissolve. In the case of membrane dissolution, we will adopt a more convenient definition as will be described in the subsequent section.

Secondly, we can add promoters and inhibitors to the rules. These are multisets of objects that regulate the rule applications in a way that the promoter $prom(r) \in \mathcal{MS}(O)$ being assigned to rule $r \in R_i$ means that $prom(r)$ must be present in m_i by any application of r. In our notation, r can only be applied if, besides the condition $lhs(r) \sqsubseteq m_i$, $prom(r) \sqsubseteq m_i$ holds provided $r \in R_i$ for some $1 \leq i \leq n$. While the inhibitor $inh(r)$, where $inh(r) \in \mathcal{MS}(O)$, prevents the rule $r \in R_i$ from being applied if any object of $inh(r)$ is present in the region i $(i \in \{1, \ldots, n\})$. That is, the rule $r \in R_i$, where $1 \leq i \leq n$, is blocked if $inh(r) \sqcap w_i \neq \emptyset$.

Finally, we make it more accurate what we mean by a membrane system with boundary rules. The fully observable base sets and observable sets are defined in the previous section. We adapt this notion to the case of P systems. The extra notion compared to P systems is the underlying approximation space together with rule applications regulated by observability. The observability condition can be considered a tool for expressing imperfect information. We apply it to P systems to impose a global context condition, allowing for powerful synchronization across the membranes as the following definition shows.

Definition 2 (*P system with boundary rules*) Let $\Pi = (O, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n)$ be a P system of order n and let $\mathfrak{B} \subseteq \mathcal{MS}(O)$ be a set of base multisets. Let us impose the following condition: a rule $u \to (v_1, here)(v_2, out)(v_3, in)$ is applicable in a configuration $C = (x_1, \ldots, x_n)$ if, in addition to the standard semantics described following Definition 1, u is observable with respect to $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$. Then Π is called a *P system with boundary rules.*

Observe that, as the membrane system evolves, the applicability of the rules may change. The primary reason is, naturally, the emergence and disappearing of the objects constituting the multisets of the regions, but, the other reason may be that the observability property of the base multisets changes. When the membrane system evolves, a

multiset can turn from being observable into non-observable and vice versa.

Now we are in a position to state our first assertion. In the proofs that follow parallel rule application is simulated by sequential one, and for this purpose a set $O' = \{a' \mid a \in O\}$ is introduced. An application of each rule of the form $u \to v$ is simulated by one or more rules transforming u to v', and only at the end of simulation of each computational step are the primed objects rewritten to the non-primed ones.

We claim that the boundary rules in sequential mode and with non-cooperative rules are enough to simulate symbol object P systems with maximally parallel mode and cooperative rules.

Theorem 1 *Let $\Pi = (O, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n)$ be a P-system of degree n with cooperative rules and maximally parallel execution mode. Then there exists a P system $\widetilde{\Pi}$ with non-cooperative, boundary rules and sequential mode such that Π and $\widetilde{\Pi}$ compute the same subsets of $\mathbb{N}$.*

Proof Let $\Pi = (O, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n)$ be a membrane system as above. We define a membrane system $\widetilde{\Pi}$ with boundary rules as follows. The main characteristic of the construction is that we separate the steps where the rule applications are simulated from the steps where the newly introduced objects are finding their place. This is accomplished by introducing primed objects for every object and, in the end of the simulation of a maximally parallel step, we exchange the primed objects for their unmarked counterparts. In the sequel, the phase where the rule applications are treated will be termed the rule application phase while the process where the primed objects are transformed into non-primed objects will be termed the communication phase.

In addition, for each object $a \in O$ so-called anti objects exist. They are denoted by $-a$. The anti-objects push the original objects into the inner part of the membrane, thus restricting their further applicability in this way. A very similar notion of anti-objects were introduced in Alhazov et al. (2014), the difference between the two notions is delicate: in our case the anti-object moves its object into the inner part of the membrane but the object is not eradicated completely. Its presence may contribute to forming the inner part and, thus, the boundary of the membrane as the P system evolves. Hence, the objects "expelled" into the inner part can still exert influence on the evolution of the P system. Therefore, the objects $-a$ only mimic the role of anti-objects, they interact with the original objects in a subtle way by means of changing the structure of the underlying multiset approximation space. Most importantly, objects and anti-object do not eliminate each other by explicit rules of the membrane system. Hence, it could be a question of discussion to what extent the emerging rules can be considered non-cooperative ones. As a shorthand, let $(-u)$ stand for $-u_1 \ldots - u_k$ if $u = u_1 \ldots u_k$.

We assume that, besides the objects O and the anti-objects, we have some designated objects, which are local or global control objects. None of them counts in the result computed by the membrane system. This approach is similar to the so-called terminal filtering, which was applied, for example, in Alhazov and Freund (2015). In our situation, the auxiliary objects do not mix with the usual objects. Either the auxiliary objects come up unchanged on both sides of a rule or the rule contains no objects other than auxiliary objects. Hence, our version of "terminal filtering" seems to be very weak.

Regarding the control objects, we reserve a separate membrane for each membrane in the newly constructed P system: if m_i is a membrane in our construction, then the control objects are placed in a membrane denoted by $m_{\underline{i}}$, except for the skin membrane. We assume that $m_{\underline{i}}$ is elementary and is a child of m_i. The skin membrane has no child of this kind. Furthermore, we assume that the skin membrane, which is membrane 1, is the output membrane both for Π and for $\widetilde{\Pi}$ and no evolution rule concerning the objects of O occurs in the skin membrane. In what follows, we present the description of the multiset approximation space.

Since the newly defined P system does not enjoy maximally parallel rule application, the auxiliary objects O' for the objects constitute a "real" part of $\widetilde{\Pi}$: they appear in the rules of $\widetilde{\Pi}$. In accordance with this, we introduce a convenient notation for the subsequent proof. Let $u \to v \in R_i$ for some $2 \le i \le n$ be a rule in m_i and assume $v_1, \ldots, v_k$ are the objects in v without the target indicators. Then, if r' is the rule corresponding to r in $\widetilde{\Pi}$, the notation v' standing on the right hand side of r' is understood as follows. We move the objects v'_j $(1 \le j \le k)$ to their respective places except for the case when $par(m_i) = skin$. In this case, the objects moving to the skin membrane are not primed. Furthermore, let us assume that the object d introduced below is in the environment, outside the skin membrane. This

supposition is not inevitable, but makes the presentation simpler. In what follows, we distinguish the membranes of the newly constructed P system from those of the old one by writing $\widetilde{m}$ instead of m. Similarly, we write $\widetilde{w}$ instead of w.

$$\begin{aligned}
O_1 =& O \cup O' \cup (-O) \cup \{\alpha, \alpha', \varepsilon, \tau, d\} \cup \\
& \{\beta_r \mid r = u \to v \in R\} \cup \\
& \{\alpha_i, \gamma_i, \gamma_i^{\spadesuit}, \gamma_i^{\clubsuit}, \varphi_i, \pi_i \mid 2 \le i \le n\}, \\
\mathfrak{B}' =& \{\alpha u, \varepsilon u \mid u \to v \in R\} \cup \\
& \{\alpha' \beta_r \alpha_i u, \alpha_i d, \tau d \mid r = u \to v \in R_i\} \cup \\
& \{\gamma_i u, \gamma_i \gamma_i, , \gamma_i^{\clubsuit} \gamma_i^{\spadesuit}, \pi_i \pi_i, \gamma_i^{\spadesuit} u, \gamma_i^{\spadesuit} d, \gamma_i u \gamma_i^{\clubsuit} \mid 2 \\
& \le i \le n, u \to v \in R_i\} \cup \\
& \{a(-a) \mid a \in O\} \cup \\
& \{\alpha' \varphi_2 \ldots \varphi_k\}
\end{aligned}$$

and

$$\begin{aligned}
\widetilde{w}_1 =& \alpha, \\
\widetilde{w}_i =& w_i \sqcup \{\beta_r \pi_i \mid r \in R_i\} \text{ if } 2 \le i \le n, \\
\widetilde{w}_{\underline{i}} =& \alpha_i \text{ if } 2 \le i \le n,
\end{aligned}$$

where the multisets $\widetilde{w}_1$, $\widetilde{w}_2$, $\widetilde{w}_{\underline{2}}, \ldots, \widetilde{w}_n, \widetilde{w}_{\underline{n}}$ denote the initial multisets in the membranes of $\widetilde{\Pi}$. We can define the sets of rules as follows:

$$\begin{aligned}
R_1' =& \{\alpha \to \alpha', \alpha' \to \varepsilon\}, \\
R_i' =& \{\beta_r \to (-u) v' \beta_r, \gamma_i^{\spadesuit} \to \varphi_i \gamma_i^{\clubsuit} \pi_i (\varepsilon_i \gamma_i, in_{\widetilde{m}_{\underline{i}}}) \mid u \to v \in R_i\} \\
& \text{provided } 2 \le i \le n, \\
R_{\underline{i}}' =& \{\alpha_i \to \gamma_i (\gamma_i^{\spadesuit} \gamma_i^{\clubsuit}, out), \gamma_i \to \tau, \tau \to \tau\} \text{ provided } 2 \le i \le n.
\end{aligned}$$

Let us give a brief explanation on how the rules operate. The example following this proof also gives some clues about the operation of the simulation. The process starts with the rule $\alpha \to \alpha'$. When the rule is applied, we know that there is an applicable rule in some of the compartments of Π by virtue of the fact that $\alpha u \in \mathfrak{B}'$ for every $r = u \to v \in R$. The fully observable base sets $\{\alpha' \beta_r \alpha_i u \mid i \in I, r \in R_i\}$ make us possible to simulate a rule application, since $\beta_r \in b^i(\widetilde{m}_i)$ if $r = u \to v \in R_i$ is applicable. Let us fix an index $2 \le i \le n$ and let the rule $r \in R_i$ be arbitrary. We recall that the skin membrane of Π does not have rules manipulating the objects in O. If $lhs(r) \sqsubseteq w_i$, then, as was mentioned above, because of $\alpha' \beta_r \alpha_i u \in \mathfrak{B}'$, β_r falls on the boundary of $\widetilde{m}_i$ and, hence, $\beta_r \to (-u) v' \beta_r$ is applicable. This involves that we simulate the changes in w_i by adding the objects denoted by v' to their respective places and, at the same time, we annihilate the objects of u from $\widetilde{w}_i$. Hence, our rules remain non-cooperative in the constructed P system $\widetilde{\Pi}$ taking into account the remark in the beginning of the proof. When there are no more rules applicable in m_i, the control element α_i evolves into γ_i in the membrane $\widetilde{m}_{\underline{i}}$ and $\gamma_i^{\spadesuit}$ and $\gamma_i^{\clubsuit}$ are emitted into $\widetilde{m}_i$. The appearance of $\gamma_i^{\spadesuit}$ and $\gamma_i^{\clubsuit}$ symbolizes the end of the rule application phase for $\widetilde{m}_i$, which is made explicit by the emergence of φ_i in $\widetilde{m}_i$ by virtue of the rule $\gamma_i^{\spadesuit} \to \varphi_i \gamma_i^{\clubsuit} \pi_i (\varepsilon_i \gamma_i, in_{\widetilde{m}_{\underline{i}}})$. The fact $\alpha' \varphi_2 \ldots \varphi_n \in \mathfrak{B}'$ synchronizes the operation of the membranes. Only if each compartment has finished its work can we shift to the communication phase by performing the rule $\alpha' \to \varepsilon$. The object π_i on the right hand side of the rule is a control object that could have remained from previous applications. Those remaining objects are pushed into the inner part of $\widetilde{m}_i$ taking into account the relations $\pi_i \pi_i \in \mathfrak{B}'$. What happens when we change from α_i to γ_i too early, that is, there are still applicable rules that have remained in m_i, yet? In this case, since $\gamma_i^{\spadesuit} u \in \mathfrak{B}'$ for any $u \in \mathcal{MS}(O)$, where $u \to v \in R_i$, the object $\gamma_i^{\spadesuit}$ cannot evolve in $\widetilde{m}_i$. Instead, by virtue of $\gamma_i u \gamma_i^{\clubsuit} \in \mathfrak{B}'$ for every $r = u \to v \in R_i$, the fact $\gamma_i u \gamma_i^{\clubsuit} \in \mathfrak{B}'$ pushes γ_i onto the boundary of $\widetilde{m}_{\underline{i}}$ and, at the same time, makes $\gamma_i u \gamma_i^{\clubsuit}$ fully observable since $u \gamma_i^{\clubsuit}$ is on the boundary of $\widetilde{m}_i$. By this, the rule $\gamma_i \to \tau$ becomes applicable, which introduces the trap object τ.

Now we demonstrate how the communication phase can be simulated. The communication phase simply means substituting every object $a' \in O'$ with $a \in O$.

$$\begin{aligned}
O_2 =& \{\lambda_i, \lambda_i' \mid 2 \le i \le n\}, \\
\mathfrak{B}'' =& \{\varepsilon a' \mid a \in O\} \\
& \cup \{\varepsilon_i d, \lambda_i a', \lambda_i' a', \lambda_i' d, \varphi_i \varphi_i, \lambda_i \lambda_i, \mid 2 \le i \le n, a \in O\} \\
& \cup \{\varepsilon \pi_2 \ldots \pi_n\},
\end{aligned}$$

where

$$\begin{aligned}
R_1'' =& \{\varepsilon \to \alpha\}, \\
R_i'' =& \{a' \to a \mid a \in O\} \\
& \cup \{\varepsilon_i \to \lambda_i (\lambda_i', out) \mid 2 \le i \le n\} \\
& \cup \{\lambda_i \to \tau, \lambda_i' \to \pi_i \varphi_i (\lambda_i, in_{\widetilde{m}_{\underline{i}}}) \mid 2 \le i \le n\}.
\end{aligned}$$

Finally, we are able to put together the components of the membrane system

$$\widetilde{\Pi} = (\widetilde{O}, \widetilde{\mu}, \widetilde{w}_1, \widetilde{w}_2, \widetilde{w}_{\underline{2}} \ldots, \widetilde{w}_n, \widetilde{w}_{\underline{n}}, \widetilde{R}_1, \widetilde{R}_2, \widetilde{R}_{\underline{2}} \ldots, \widetilde{R}_n, \widetilde{R}_{\underline{n}})$$

with boundary rules and underlying multiset approximation space $\mathfrak{B}$, where

$$\widetilde{O} = O_1 \cup O_2,$$
$$\mathfrak{B} = \mathfrak{B}' \cup \mathfrak{B}'',$$
$$\widetilde{R}_1 = R'_1 \cup R''_1,$$
$$\widetilde{R}_i = R'_i \cup R''_i,$$
$$\widetilde{R}_{\underline{i}} = R'_{\underline{i}} \quad (2 \le i \le n).$$

Moreover, $\widetilde{\mu}$ is the same as μ, except for the fact that every $\widetilde{m}_i \in \mu$ that is not the *skin* membrane is augmented with an additional child membrane $m_{\underline{i}}$, which is an elementary membrane $(2 \le i \le n)$. The multisets $\widetilde{w}_1, \widetilde{w}_2, \widetilde{w}_{\underline{2}} \ldots, \widetilde{w}_n, \widetilde{w}_{\underline{n}}$ were defined above.

Let us give an informal explanation of how the communication step is simulated. As it was remarked before, the communication phase models the placement of the objects on the right hand side of the rules into their respective membranes. The start of the communication phase is indicated by the appearance of ε in the skin membrane. Then, by the fact $\varepsilon a' \in \mathfrak{B}''$ $(a' \in O')$, the rule $a' \to a$ is applicable. As before, let $2 \le i \le n$ be arbitrary. The objects a' are transformed back to the objects a at this stage. When there are no more $a' \in \widetilde{w}_i$, then $\varepsilon_i \to \lambda_i(\lambda'_i, out)$ is applied. The rules $\lambda'_i \to \pi_i \varphi_i(\lambda_i, in_{\widetilde{m}_{\underline{i}}})$ are triggered by the fact $\lambda'_i d \in \mathfrak{B}''$, which make in the end the application of $\varepsilon \to \alpha$ possible by reason of the fact $\varepsilon\pi_2 \ldots \pi_n \in \mathfrak{B}''$. What can be said when the objects λ_i, λ'_i are introduced too early and there are still a' in $\widetilde{m}_i$? Assume $a' \in \widetilde{w}_i$ for some $a' \in O'$. In this case, the object λ'_i is stuck in the inner part of $\widetilde{m}_i$, that is, in $l(\widetilde{m}_i)$, however, the rule $\lambda_i \to \tau$ can be applied by virtue of $\lambda_i a' \in \mathfrak{B}''$. But this introduces the trap object τ. Hence, the computation remains meaningful only if, in each membrane, the point when the elements a' are eliminated is guessed correctly. In this case, α appears in the skin membrane and a new computational step can be simulated. □

Let us look at an example, to see how a P system with maximal parallel execution mode is simulated in our construction. Our example contains only one additional membrane besides the skin membrane, since the intention is to demonstrate how the maximal parallel step is actually simulated together with keeping the membrane systems as simple as possible to facilitate reading.

Example 2 Let $\Pi = (O, \mu, w_1, w_2, R_1, R_2)$ be a P-system, where

$$w_2 = a^3 b^2 c,$$
$$R_2 = \{r_{21} : ab \to a(c, out), r_{22} : a^2 c \to b\}.$$

moreover, let us assume that w_1 is the null multiset and R_1 is empty. We simulate the maximally parallel step that consists of the applications of the rules r_{21}, r_{22} in R_2. Initially, $\widetilde{\Pi} = (O \cup O', \widetilde{\mu}, \widetilde{w}_1, \widetilde{w}_2, \widetilde{w}_{\underline{2}}, R'_1 \cup R''_1, R'_2 \cup R''_2, R'_{\underline{2}})$, where

$$\widetilde{\mu} = [\,[\,[\,]_{\underline{2}}\,]_2\,]_1,$$
$$\widetilde{w}_1 = \alpha,$$
$$\widetilde{w}_2 = a^3 b^2 c \sqcup \beta_{r_{21}} \beta_{r_{22}} \pi_2,$$
$$\widetilde{w}_{\underline{2}} = \alpha_2,$$

In addition,

$$R'_1 = \{\alpha \to \alpha', \alpha' \to \varepsilon\},$$
$$R'_2 = \{r'_{21} : \beta_{r_{21}} \to (-ab)a'(c, out)\beta_{r_{21}}, r'_{22} : \beta_{r_{22}} \to (-a^2 c)b'\beta_{r_{22}}, \gamma_2^{\spadesuit} \to \varphi_2 \gamma_2^{\clubsuit} \pi_2(\varepsilon_2 \gamma_2, in_{\widetilde{m}_{\underline{2}}})\},$$
$$R'_{\underline{2}} = \{\alpha_2 \to \gamma_2(\gamma_2^{\spadesuit}\gamma_2^{\clubsuit}, out), \gamma_2 \to \tau, \tau \to \tau\},$$
$$R''_1 = \{\varepsilon \to \alpha\},$$
$$R''_2 = \{a' \to a \mid a \in O\} \cup \{\varepsilon_2 \to \lambda_2(\lambda'_2, out)\} \cup \{\lambda_2 \to \tau, \lambda'_2 \to \pi_2 \varphi_2(\lambda_2, in_{\widetilde{m}_{\underline{2}}})\},$$

Now we define the base sets for the approximation space:

$$\mathfrak{B}' = \{\alpha ab, \alpha a^2 c, \varepsilon ab, \varepsilon a^2 c\} \cup \{\alpha' \beta_{r_{21}} \alpha_2 ab, \alpha' \beta_{r_{22}} \alpha_2 a^2 c, \alpha_2 d, \tau d\} \cup \{\gamma_2 ab, \gamma_2 a^2 c, \gamma_2\gamma_2, \gamma_2^{\clubsuit}\gamma_2^{\clubsuit}, \pi_2\pi_2, \gamma_2^{\spadesuit} ab, \gamma_2^{\spadesuit} a^2 c, \gamma_2^{\spadesuit} d, \gamma_2 ab \gamma_2^{\clubsuit}, \gamma_2 a^2 c \gamma_2^{\clubsuit}\} \cup \{a(-a) \mid a \in O\} \cup \{\alpha' \varphi_2\},$$
$$\mathfrak{B}'' = \{\varepsilon a' \mid a \in O\} \cup \{\varepsilon_2 d, \lambda_2 a', \lambda'_2 a', \lambda'_2 d, \varphi_2\varphi_2, \lambda_2\lambda_2 \mid a \in O\} \cup \{\varepsilon\pi_2\}.$$

We follow some of the computational steps. Firstly, let us examine what happens in membrane 2 when rules r_{21} and r_{22} are applied in the maximally parallel step. Since $ab \sqsubseteq w_2$ and $\alpha ab \in \mathfrak{B}'$, the object α is on the boundary of $\widetilde{m}_1$ and αab is fully observable, hence, $\alpha \to \alpha'$ is applicable. This is the only rule that can be applied in $\widetilde{\Pi}$ at that step. Now, α' is introduced in $\widetilde{m}_1$. Since $\alpha' \beta_{r_{21}} \alpha_i ab \in \mathfrak{B}'$, and $\alpha' \beta_{r_{21}} \alpha_i ab$ is fully observable, then $\beta_{r_{21}}$ is on the observable boundary of w_2. Thus we can apply $\beta_{r_{21}} \to (-ab)a'(c, out)\beta_{r_{21}}$. Continuing in this spirit, we obtain the following reduction sequence:

$$a^3 b^2 c \beta_{r_{21}} \beta_{r_{22}} \pi_2 \to (-ab)a'(c, out) a^3 b^2 c \beta_{r_{21}} \beta_{r_{22}} \pi_2 \to (-a^3 bc)a'b'(c, out) a^3 b^2 c \beta_{r_{21}} \beta_{r_{22}} \pi_2.$$

If we perform $\alpha_2 \to \gamma_2(\gamma_2^{\spadesuit}\gamma_2^{\clubsuit}, out)$ in $\widetilde{m}_{\underline{2}}$ next, then γ_2 will appear in $\widetilde{w}_{\underline{2}}$ and $\gamma_2^{\spadesuit}$ and $\gamma_2^{\clubsuit}$ will be introduced to $\widetilde{m}_2$. Now, since $(-a^3bc)a^3bc \sqsubseteq l(\widetilde{m}_2)$, none of the base sets in which γ_2 appears can be fully observable. Hence, the rule $\gamma_2 \to \tau$ is suppressed. On the other hand, $\gamma_2^{\spadesuit}$ can evolve because of $\gamma_2^{\spadesuit} d \in \mathfrak{B}'$. This results in $\widetilde{w}_2 = (-a^3bc)a'b'a^3b^2c\beta_{r_{21}}$ $\beta_{r_{22}}\pi_2\pi_2\gamma_2^{\clubsuit}\gamma_2^{\clubsuit}\phi_2$. Observe that c is not an element of $\widetilde{w}_2$. By the same rule, a copy of γ_2 is moved into $\widetilde{m}_{\underline{2}}$. This means that α' falls on the observable boundary of $\widetilde{m}_1$ and, hence, $\alpha' \to \varepsilon$ is applicable. The erasure of the possibly emerging objects a', b' is the subsequent step of the simulation in order to return to the next computational step of the computation.

We examine now the case when the simulation of a maximally parallel step would end prematurely. Assume that, in the previous process, we have simulated rule r_{21} correctly, but, instead of continuing with the simulation of r_{22} we abandon that sequence of rule applications. This means that we apply rule $\alpha_2 \to \gamma_2(\gamma_2^{\spadesuit}\gamma_2^{\clubsuit}, out)$ in $\widetilde{m}_{\underline{2}}$ as the next step following r'_{21}. Then the configuration $(-ab)a'a^3b^2c\beta_r\pi_2\gamma_2^{\spadesuit}\gamma_2^{\clubsuit}$ arises in $\widetilde{m}_2$. Now, since α_2 is replaced by γ_2 in $\widetilde{m}_{\underline{2}}$ and $a^2c \in b^i(\widetilde{m}_2)$, the base set $\gamma_2 a^2 c$ is fully observable and the rule $\gamma_2 \to \tau$ can be applied, which introduces the trap object τ. On the other hand, since $\gamma_2^{\spadesuit} a^2 c \in \mathfrak{B}'$, the object $\gamma_2^{\spadesuit}$ is sent into the inner part of $\widetilde{m}_2$, while the fact $\gamma_2 a^2 c \gamma_2^{\clubsuit} \in \mathfrak{B}'$ ensures that a^2c remains on the boundary of $\widetilde{m}_2$ preserving the observability of γ_2.

4 Boundary rules and auxiliary membrane computational features

In this section, it is investigated how boundary rules can replace both membrane dissolution and promoter/inhibitor rules. In the sequel, to make matters simpler, we assume the presence of the maximally parallel execution mode. However, it should be clear by the above discussion that after having eliminated the membrane dissolution or the promoter/inhibitor rules we are able to eliminate the remaining maximally parallel execution and simulate the P system with non-cooperative rules in sequential mode.

Let us begin our discussion with membrane dissolution. We assume maximally parallel execution mode and we prove that boundary rules are enough to simulate membrane dissolution. We recall that the symbol δ marks a region for dissolution. When it is introduced in the membrane by an application of a rule, then, after the maximally parallel and communication steps, the actual membrane containing δ disappears. Its objects, except for δ, move to the parent membrane and its rules cannot be applied anymore.

Let us choose a rather convenient way of interpreting the behaviour of the membrane system in the presence of the object δ: we do not dissolve the membranes where δ is introduced. However, we implement the synchronization demanded by the δ object as follows: firstly, the rule applications are performed together with the communication phase. A timing construct ensures that these processes finish at the same time in each membrane. Then we check whether δ is present in the membrane. In this case, the necessary redirection of the objects into the parent membrane is performed. The process must be recursive: if the parent membrane of a membrane already contains δ, then the objects must find its parent membrane and so on. Let us choose the easier way here: instead of a bottom-up traversal of the tree let us iterate this process sufficiently many times in order to ensure that the placement of the objects finishes.

As before, new sets of objects, "anti-objects", are introduced: if a is any object, the pair a, $-a$ extinguish each other, or, better to say, an object a is neutralized by the introduction of the anti-object $-a$, however, they can affect the further evolution of the P system as was discussed in the proof of Theorem 1. In effect, this means that $a(-a)$ moves into the inner part of the membrane. This is the tool that ensures that every membrane of our constructed P system, $\widetilde{\Pi}$, should contain the correct number of objects when it turns to simulation of rules of Π. As before, we write $(-u)$ for $-u_1 \ldots - u_k$ if $u = u_1 \ldots u_k$.

Theorem 2 *Let $\Pi = (O, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n)$ be a symbol object P system of degree n with membrane dissolution and maximally parallel execution mode. Then there exists a P system $\widetilde{\Pi}$ with non-cooperative boundary rules and sequential mode such that Π and $\widetilde{\Pi}$ compute the same subsets of $\mathbb{N}$.*

Proof Let $\Pi = (O, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n)$ be a symbol object P system of degree n with membrane dissolution. As we have discussed before, it is enough to show that membrane dissolution can be simulated with non-cooperative boundary rules in the maximally parallel mode, since the maximally parallel execution mode can then be replaced by a sequential execution mode by Theorem 1. Since we return to the maximally parallel execution mode even by the simulating P systems, in the following proof we omit dealing with primed objects. The objects directly move to their appropriate places during the rule applications and the system simply "knows" when a maximal multiset of rules has been chosen. When the maximally parallel rule application is finished, we have to ensure the correct placement of the objects but only in connection with δ. The novelty, compared to the previous section, is that the rearrangement of the objects of the P system due to

the presence of δ must also be translated in terms of the boundary rules. Having finished the rule applications, the objects of a compartment where δ is present must move to the parent membrane and this process is continued till each compartment containing a δ is emptied (except for the δ). The next computational step can only take place after having removed every object from the membranes with δ, however, δ is not removed from a membrane. As before, we denote the membranes of $\widetilde{\Pi}$ by $\widetilde{m}$ instead of m and the configurations by $\widetilde{w}$ instead of w, respectively.

Turning to the simulation itself, the appearance of object α in the skin membrane will indicate the start of the simulation of a new computational step. The first phase of a computational step is termed the rule application phase and the phase where the objects find their membranes moving through the membranes marked for disappearance by δ will be called the δ phase. The simulation of the δ phase starts with the introduction of ε in the skin membrane. However, there is a problem that must be addressed appropriately. Since the objects of O are manipulated not only in the rule application phase but also in the δ phase, the entanglement of the two kinds of applications should be avoided. We resolve this by creating "second copies" of the original membrane contents. If $a \in \widetilde{w}_i$ for any membrane $\widetilde{m}_i$ $(1 \le i \le n)$ and any object $a \in O$, then the corresponding object $a^{\#}$ will also appear in $\widetilde{m}_i$. Regarding the construction below, first of all it is described how $\widetilde{\Pi}$ simulates the rule application part of a computational step of Π. This part is very similar to the proof of Theorem 1 owing to our effort to remain by a non-cooperative membrane system. The presence of the maximally parallel mode simplifies the construction a bit, however. Then we discuss how membrane dissolution is handled. It is assumed that there are no evolution rules concerning the objects of O in the skin membrane, which is membrane $\widetilde{m}_1$. Moreover, for the sake of simplicity, let us assume that $\widetilde{m}_1 = par(\widetilde{m}_2)$. In the following construction, the initial objects and rules for membrane 2 will differ from those of the rest of the membranes. Additionally, as in Theorem 1, we assume that only objects of O are transported into the skin membrane from its child membranes. We do not indicate this by the formulation of the rules to keep the presentation uniform. This does not mean any restriction, however, since the skin membrane does not contain evolutionary rules for the objects of O. Let $\widetilde{\Pi} = (\widetilde{O}, \widetilde{\mu}, \widetilde{w}_1, \ldots, \widetilde{w}_n, \widetilde{R}_1, \ldots, \widetilde{R}_n)$ be defined as follows.

$$
\begin{aligned}
\widetilde{O} &= O \cup O^{\#} \cup -O \cup -O^{\#} \cup \{\delta\} \cup \{p_r \mid r = u \to v \in R\} \\
&\cup \{\alpha, \alpha', \varepsilon, \varepsilon', \varepsilon'', \gamma, \gamma', \theta, \theta', \mu, \nu, \tau, \phi, \psi\}, \\
\mathfrak{B} &= \{\alpha u \mid u \in \mathcal{MS}(O), u \to v \in R_i, \widetilde{m}_i \neq skin\} \\
&\cup \{p_r u \alpha' \mid u \in \mathcal{MS}(O), u \to v \in R_i, \widetilde{m}_i \neq skin\} \\
&\cup \{\gamma\theta'\mu, \gamma'\theta'\nu\} \cup \{\gamma u \mid r = u \to v \in R\} \\
&\cup \{a(-a) \mid a \in O\} \\
&\cup \{a^{\#}(-a^{\#}) \mid a \in O\} \\
&\cup \{\varepsilon'\theta'\phi, \varepsilon''\theta'\psi\} \\
&\cup \{\varepsilon a^{\#}\delta, \varepsilon' a\delta \mid a \in O\} \\
&\cup \{\alpha' d, \varepsilon d, \theta d, \tau d, \gamma\gamma, \gamma'\gamma', \varepsilon''\varepsilon'', \theta'\theta'\},
\end{aligned}
$$

and

$$
\begin{aligned}
\widetilde{w_1} &= \alpha, \\
\widetilde{w_2} &= w_2 w_2^{\#} \sqcup \mu\nu\phi\psi \sqcup \{p_r \mid r \in R_i\}, \\
\widetilde{w_i} &= w_i w^{\#} \sqcup \{p_r \mid \in R_i\}_i \quad (3 \le i \le n).
\end{aligned}
$$

Letalso

$$
\widetilde{R} = \widetilde{R}_1 \bigcup_{i=2}^{n} \widetilde{R}_i,
$$

with $\widetilde{R}_i = \widetilde{R}_i^1 \cup \widetilde{R}_i^2$, where

$$
\begin{aligned}
\widetilde{R}_1 &= \{\alpha \to \alpha', \alpha' \to \gamma\theta, \gamma \to \gamma', \varepsilon \to \varepsilon'\theta, \theta \to \theta'\}, \\
\widetilde{R}_2^1 &= \{\mu \to (\varepsilon\gamma'\theta', out), \nu \to \tau, \psi \to \tau, \tau \to \tau\} \\
&\quad \cup \{p_r \to vv^{\#} - (uu^{\#})p_r \mid u \to v \in R_2\}, \\
\widetilde{R}_2^2 &= \{a^{\#} \to (-a)(aa^{\#}, out), \varepsilon' \to \varepsilon'', \phi \to (\alpha\varepsilon''\theta', out), \psi \to \tau\}, \\
\widetilde{R}_i^1 &= \{p_r \to vv^{\#} - (uu^{\#})p_r \mid u \to v \in R_i\} \quad \text{and} \\
\widetilde{R}_i^2 &= \{a^{\#} \to -a(aa^{\#}, out)\} \quad (3 \le i \le n).
\end{aligned}
$$

The structure $\widetilde{\mu}$ of the P system is the same as that of Π. First of all, let us recall that $\widetilde{m}_1$ is supposed to be the skin membrane. Let d be an object in the "environment" outside of $\widetilde{m}_1$. In what follows, the simulation is described in an informal way. As mentioned above, two copies of the objects are created, the original ones being elements of O while the copies are elements of $O^{\#}$. The rule applications are simulated in membrane $\widetilde{m}_i$ $(2 \le i \le n)$ by the rules $\widetilde{R_i^1}$, while the rules $\widetilde{R_i^2}$ serve for the placement of the objects in connection with the appearance of δ. Let $2 \le i \le n$ be arbitrary. We discuss the case of the rule applications first. The rule $\alpha \to \alpha'$ in $\widetilde{m}_1$ initiates the computational step if α falls on the boundary of $\widetilde{m}_1$ by virtue of $\alpha u \in \mathfrak{B}$ for every $r = u \to v \in R$. If there are applicable rules, then the objects p_r will lie on the boundary of $\widetilde{m}_i$. The rules of R_i are simulated by the non-cooperative rules $p_r \to vv^{\#} - (uu^{\#})p_r$. This means that we add the objects of $v^{\#}$ and those of the original right hand side v to $\widetilde{m}_i$ and we also subtract the objects in u and $u^{\#}$

from $\widetilde{m}_i$ by introducing their "anti-objects". We continue in this way until there are no more applicable rules in m_i. We try to guess when the rule application phase is over by preforming the rule $\alpha' \to \gamma\theta$ in $\widetilde{m}_1$. If the rule was applied in the right time, then θ changes into θ' but γ cannot evolve. In the next time instance, owing to the fact $\gamma\tau'\mu \in \mathfrak{B}$, μ evolves and $\varepsilon\gamma'\theta'$ is placed in membrane 1 and, at the same time, γ is transformed into γ' (since we are in maximally parallel execution mode), which means the annihilation of γ' and θ' in $\widetilde{m}_1$ by expelling them from the boundary to the inner part and the δ phase can start. Otherwise, if $r = u \to v \in R$ would still be applicable in the presence of γ, then, by reason of $\gamma u \in \mathfrak{B}$, γ evolves to γ' and, since $\gamma'\theta' v \in \mathfrak{B}$, v creates τ in membrane 2, which is a trap object.

The implementation of the dissolution part is started with the introduction of ε in $\widetilde{m}_1$. By virtue of the facts $\varepsilon a^{\#}\delta \in \mathfrak{B}$, the rule $a^{\#} \to (-a)(aa^{\#}, out)$ becomes applicable when δ is present in the membrane. This mimics expelling an object a into the parent membrane: in our case we have to move its "twin" object simultaneously. Finally, when the placement of the objects finishes, the rule $\varepsilon \to \varepsilon'$ is applied. If the rule was applied correctly, then, by $\varepsilon'\theta'\phi \in \mathfrak{B}$, $\phi \to (\alpha\varepsilon''\theta', out)$ can be performed and a new computational step can be simulated. (ε' evolves to ε'' in the same time.) Otherwise, by reason of $\varepsilon' a\delta \in \mathfrak{B}$ $(a \in O)$ and the maximally parallel execution mode, ε' and θ can evolve simultaneously. Then, because of $\varepsilon''\theta'\psi \in \mathfrak{B}$, ψ falls on the boundary of $\widetilde{m}_2$, and, hence, introduces the trap object τ. □

Finally, the simulation of rules with promoter and inhibitor is discussed. We recap again that, in the presence of promoters and inhibitors, we assign to each rule $r \in \mathcal{R}$ the so called promoter/inhibitor multisets ($prom(r)/inh(r)$), which fall under the conditions that r can be applied to w_i only if, besides the usual requirement $lhs(r) \sqsubseteq w_i$, every object of $prom(r)$ is present in w_i, that is, $prom(r) \sqsubseteq w_i$ and, in the same time, no element of $inh(r)$ can be found in w_i. In other words, $inh(r) \sqcap w_i = \emptyset$. The basic idea of the construction is simple: if the multiset $prom(r)$ is present in the membrane, then we add an object p_r to the membrane such that the execution of rule r becomes enabled. In case an inhibitor object is present, we move p_r into the inner part of the membrane thus preventing the application of the membrane rule. The only difficulty emerging is that we must check whether $prom(r) \sqsubseteq w_i$ and $inh(r) \sqcap w_i \neq \emptyset$ for every membrane m_i and every rule $r \in R_i$ $(1 \leq i \leq n)$ before applying the rule. Since the promoters and inhibitors can compete for the membrane objects, we can only do this one by one for each rule separately. Let us assume that every membrane contains exactly s rules (we apply dummy rules, if necessary). Having finished with the examination of all the rules, the simulation of the actual rule applications can begin. In order to uniform rule applications in the membranes of $\widetilde{\Pi}$, we also assume that each promoter is of exactly the same length. The key idea is that the checking for the promoters and inhibitors should finish in the same time in each membrane and, thus, we ensure a synchronization for the whole P system. Below, we describe the approximation space together with the rules.

Theorem 3 *Let $\Pi = (O, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n)$ be a symbol object P system of degree n with promoters/inhibitors and maximally parallel execution mode. Then there exists a P system $\widetilde{\Pi}$ with non-cooperative boundary rules and sequential mode such that Π and $\widetilde{\Pi}$ compute the same subsets of $\mathbb{N}$.*

Proof Let $\Pi = (O, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n)$ be a symbol object P system of degree n with promoters/inhibitors. As before, it is proven that the promoter/inhibitor construction can be eliminated with the help of non-cooperative boundary rules when the the maximally parallel execution mode is applied. From the above constructions it is clear that, having removed promoters/inhibitors from the P system, we are able to substitute the maximally parallel execution mode with a sequential mode following the argumentation of Theorem 1. As before, two copies of the object set is maintained, this time the second copy is denoted by $\underline{O}$. We also have the set of "anti-objects" both for O and $\underline{O}$. For notational convenience, let us assume that there are as many as s rules in each compartment and the promoters have equal lengths for each rule (we use dummy objects if necessary). Similarly to the proof of Theorem 1, a child membrane $m_{\underline{i}}$, which is an elementary membrane, is created for each membrane m_i, except for the skin membrane. The auxiliary objects not underlined are usually belong to membranes $m_{\underline{i}}$, while their underlined counterparts are kept or created in the original membrane m_i. In the sequel we use the notation $prom(r) = prom(r, 1)\ldots prom(r, d)$ for the promoter of the rule r, where $d = |prom(r)|$ the common length of the promoters. The skin membrane does not have rules for the evolution of the objects in O and, in the course of the computation, it does not contain objects except for t and the objects forming the result. The equations below define the membrane system $\widetilde{\Pi} = (\widetilde{O}, \widetilde{\mu}, \widetilde{m}_1, \widetilde{m}_2, \widetilde{m}_{\underline{2}}, \ldots, \widetilde{m}_n, \widetilde{m}_{\underline{n}}, \widetilde{R}_1, \widetilde{R}_2, \widetilde{R}_{\underline{2}}, \ldots, \widetilde{R}_n, \widetilde{R}_{\underline{n}})$.

$$
\begin{aligned}
\widetilde{O} =& O \cup -O \cup \underline{O} \cup -\underline{O} \cup \{p_r^0, p_r, q_r^0, q_r \mid r \in R\} \\
& \cup \{\alpha_i^k, \underline{\alpha_i^k}, \underline{\underline{\alpha_i^k}}, \beta_i \mid 1 \le i \le s, 1 \le k \le d = |prom(r_i)|\} \\
& \cup \{\gamma_1, \gamma_2, t\}, \\
\mathfrak{B} =& \{\alpha_i^k \, prom(r_i, k) \mid 1 \le i \le s, 1 \le k \le d\} \\
& \cup \{\underline{\alpha_i^k} \, \underline{prom(r_i, k)} \mid 1 \le i \le s, 1 \le k \le d\} \\
& \cup \{\beta_i a \mid 1 \le i \le s, a \in inh(r_i)\} \\
& \cup \{\underline{\beta_i} a \mid 1 \le i \le s, a \in inh(r_i)\} \\
& \cup \{\underline{\alpha_i^k} t, \underline{\underline{\alpha_i^k}} t \mid 1 \le i \le s, 1 \le k \le d\} \\
& \cup \{\gamma_1 \underline{a} \mid a \in O\} \\
& \cup \{up_{r_i}^0 t \mid r_i = u \to v \in R_j, 1 \le i \le s\} \\
& \cup \{up_{r_i} t \mid r_i = u \to v \in R_i, 1 \le i \le s\} \\
& \cup \{\gamma_1 p_{r_i}^0 t, \gamma_1 q_{r_i}^0 t, \gamma_2 t \mid 1 \le i \le s\} \\
& \cup \{p_{r_i} q_{r_i} \mid 1 \le i \le s\}, \text{ and} \\
\widetilde{w}_1 =& t, \\
\widetilde{w}_i =& w_i \underline{w_i} \, \underline{\alpha_1^1}, \\
\widetilde{w}_{\underline{i}} =& \alpha_1^1 \quad (2 \le i \le n)
\end{aligned}
$$

Inwhat follows, we present the set of rules $\widetilde{R}_k = \widetilde{R}_k' \cup \widetilde{R}_k''$ for membrane $\widetilde{m}_k$ and the set of rules $\widetilde{R}_{\underline{k}}$ for $\widetilde{m}_{\underline{k}}$, where $\widetilde{m}_k$ is a fixed membrane different from the skin membrane and the set of rules $\widetilde{R}_1$ for the skin membrane. We obtain $\widetilde{R}$ if we let $\widetilde{R} = \widetilde{R}_1 \bigcup \cup_{k=2}^{n} (\widetilde{R}_k' \cup \widetilde{R}_k'' \cup \widetilde{R}_{\underline{k}})$. Let $2 \le k \le n$ bearbitrary.

$$
\begin{aligned}
\widetilde{R}_1 =& \emptyset, \\
\widetilde{R}_k' =& \{\alpha_i^j \to \alpha_i^{j+1}(\underline{\underline{prom(i,j)}}(-\underline{\alpha_i^j prom(i,j)}) \, \underline{\alpha_i^{j+1}}, out) \mid \\
& 1 \le i \le s, 1 \le j < d = |prom(r_i)|\} \\
& \cup \{\alpha_i^d \to (p_{r_i}^0, out)\alpha_{i+1}^1(\underline{\underline{prom(i,d)}} - prom(i,d)(\underline{-\alpha_i^d}) \, \underline{\alpha_{i+1}^1}, out) \mid \\
& d = |prom(r_i)|, 1 \le i < s\} \\
& \cup \{\underline{\alpha_i^j} \to \underline{\underline{\alpha_i^{j+1}}}(-\alpha_i^j, in_{\widetilde{m}_{\underline{k}}}) \mid 1 \le i < s, 1 \le j \le d\} \\
& \cup \{\underline{\underline{\alpha_i^j}} \to \underline{\underline{\alpha_i^{j+1}}} \mid 1 \le j \le d\} \\
& \cup \{\underline{\alpha_i^d} \to \underline{\alpha_{i+1}^1}(\alpha_{i+1}^1, in_{\widetilde{m}_{\underline{k}}}) \mid 1 \le i < s\} \\
& \cup \{\underline{\underline{\alpha_i^d}} \to \underline{\alpha_{i+1}^1}(\alpha_{i+1}^1, in_{\widetilde{m}_{\underline{k}}}) \mid 1 \le i < s\} \\
& \cup \{\alpha_s^d \to \beta_1(p_{r_s}^0 \, \underline{\underline{prom(s,d)}} - prom(s,d) - \underline{\alpha_s^d}) \, \underline{\beta_1}), out\} \\
& \cup \{\underline{\alpha_s^d} \to \underline{\beta_1}(-\alpha_s^d \beta_1, in_{\widetilde{m}_{\underline{k}}})\} \\
& \cup \{\underline{\underline{\alpha_s^d}} \to \underline{\beta_1}(\beta_1, in_{\widetilde{m}_{\underline{k}}})\} \\
& \cup \{\beta_i \to \beta_{i+1}(q_{r_i}^0(-\underline{\beta_i}) \, \underline{\beta_{i+1}}, out), \\
& \underline{\beta_i} \to \underline{\beta_{i+1}}(\beta_{i+1}, in_{\widetilde{m}_{\underline{k}}}) \mid 1 \le i < s\} \\
& \cup \{\underline{\underline{a}} \to a \mid a \in O\} \\
& \cup \{\beta_s \to \gamma_1(-\underline{\beta_s}, out)\}
\end{aligned}
$$

and

$$
\begin{aligned}
\widetilde{R}_m'' =& \{p_{r_i} \to (-u)v(-\underline{u})\underline{v} \mid r = u \to v \in R_j\} \\
& \cup \{p_{r_i}^0 \to p_{r_i}, q_{r_i}^0 \to q_{r_i} \mid 1 \le i \le s\}, \\
\overline{R}_m =& \{\gamma_1 \to \gamma_2, \gamma_2 \to \alpha_1^1(\underline{\alpha_1^1}, out)\}.
\end{aligned}
$$

Again,an informal explanation is presented. Prior to every rule application of Π, let us collect the rules that are allowed by their promoters or prohibited by their inhibitors in that computational step. Firstly, we check whether $prom(r_i) \sqsubseteq w_j$ if $r_i \in R_j$. This is done as follows. As above, we write $prom(r_i) = prom(r_i, 1)\ldots prom(r_i, d)$ with $prom(r_i, l) \in O$ and $d = |prom(r_i)|$. We examine in each membrane separately whether the conditions imposed by $prom(r_i)$ and $inh(r_i)$ are fulfilled. In order to maintain synchronization, each membrane is supposed to contain exactly as many as s rules, hence the process finishes at the same time everywhere in the P system. Let $2 \le i \le n$ be arbitrary. The process starts by checking rule 1 of R_i, which we denote by r_1, with the help of α_1^1. Note that there is a slight impreciseness here: we do not indicate the fact that $r_1 \in R_i$ in the notation used for the αs and βs in order to not overburden our notation and we do not show that $r_1 \in R_i$ in the indexing of the rule, either. By the stipulation $\alpha_1^1 \, prom(r_1, 1) \in \mathfrak{B}$ the rule $\alpha_1^1 \to \alpha_1^2 \, (\underline{\underline{prom(r_1, 1)}} - (\underline{\alpha_1^1} \, prom(r_1, 1)) \, \underline{\alpha_1^2}, out)$ can always be applied provided $prom(r_1) \sqsubseteq w_i$. This means that α_1^2 appears in $\widetilde{m}_{\underline{i}}$ and $\underline{\alpha_1^1}$ is neutralized by $-\underline{\alpha_1^1}$ in $\widetilde{m}_i$ and $\underline{\alpha_1^2}$ is also sent out to $\widetilde{m}_i$. At the same time, a copy of $prom(r_1, 1)$ is erased from $\widetilde{m}_i$ by driving $-prom(r_1, 1)$ into $\widetilde{m}_i$. The auxiliary element $\underline{\underline{prom(r_1, 1)}}$ is placed in $\widetilde{m}_i$ in order to enable us to restore $prom(r_1)$ when necessary. On the other hand, if $prom(r_1) \sqsubseteq \widetilde{w}_i$ does not hold and it turns out because of the fact $prom(r_1, 1) \notin \widetilde{w}_i$, then $\underline{\alpha_1^1}$ falls at the boundary of $\widetilde{m}_i$ due to $\underline{\alpha_1^1} t \in \mathfrak{B}$ and the rule $\underline{\alpha_1^1} \to \underline{\underline{\alpha_1^2}}(-\alpha_1^1, in_{\widetilde{m}_{\underline{i}}})$ applies. If the object $\underline{\underline{\alpha_1^2}}$ is introduced, we never return to the objects α_1^p, hence $p_{r_1}^0$ will not appear in $\widetilde{m}_i$. We iterate the process for the other rules of R_i until we reach β_1. If $prom(r_j) \sqsubseteq \widetilde{w}_i$ was indeed the case, then $p_{r_j}^0$ is added to $\widetilde{m}_i$. Otherwise, we switch to the objects $\underline{\alpha_j^k}$ and $\underline{\underline{\alpha_j^k}}$ until we reach α_{j+1}^1 and $p_{r_j}^0$ will not be introduced. The objects $p_{r_j}^0$ indicate that rule r_j satisfies the requirement imposed by $prom(r_j)$. Otherwise, if an element of the multiset $inh(r_j)$ occurs in $\widetilde{w}_i$ provided $r_j \in R_i$, then r_j is inhibited. In this case, the object $q_{r_j}^0$ is added to $\widetilde{w}_i$. We check this for every rule one by one going through the objects β_j $(1 \le j \le s)$. This time each β_j is connected with a rule, namely $r_j \in R_i$. The overall process is finished when we reach β_s. There is one administrative task left: we checked the correct simulation of the multisets $prom(r_j)$ by eliminating in membrane $\widetilde{w}_i$ the already found elements $prom(r_j, l)$ and, in the same time, we sent an object $\underline{\underline{prom(r_j, l)}}$ into $\widetilde{w}_i$. The created objects $\underline{\underline{prom(r_j, l)}}$ should be transformed back to objects $prom(r_j, l)$ in order

to give back to $\widetilde{w}_i$ the missing elements. This is done in the presence of γ_1 and in one maximally parallel step by the rules $\underline{\underline{a}} \to a$ $(a \in O)$.

Lastly, the simulation of the rule applications of Π can start with the introduction of γ_1. Termination is ensured by the conditions $up^0_{r_i} t \in \mathfrak{B}$, where $r_i = u \to v \in R_k$ for some $2 \le k \le n$ and $1 \le i \le s$. Only when at least one of them is fulfilled can the process be continued. Then the objects $p^0_{r_i}$ and $q^0_{r_i}$ evolve to p_{r_i} and q_{r_i}, respectively. When p_{r_i} is present, the rule $r_i = u \to v \in R_k$ can be simulated by virtue of the fact $up_{r_i} t \in \mathfrak{B}$. The appearance of q_{r_i}, however, blocks the rule application, since $p_{r_i} q_{r_i} \in \mathfrak{B}$. At the same time, γ_2 is introduced and this leads back to the beginning of the process since γ_2 creates α^1_1 and $\underline{\alpha^1_1}$. □

5 Conclusion

In this paper, P systems equipped with an underlying multiset approximation space were studied. The idea grew out from the works of Pawlak concerning rough sets as a way of reasoning about data in artificial intelligence (Pawlak 1982, 1991), and was further pursued in Mihálydeák and Csajbók (2013, 2014), Mihálydeák and Vaszil (2015), Battyányi et al. (2019) and Battyányi and Vaszil (2019). Symbol object membrane systems were considered with the maximally parallel execution mode and it has been shown that the dynamic rule applicability property ensured by the underlying multiset approximation space was enough for simulating the computational properties of the original membrane system by a membrane system with multiset approximation space in sequential mode and operating only with non-cooperative rules. Furthermore, it was demonstrated that the membrane dissolution extension and the presence of promoter/inhibitor rules could be simulated with P systems with appropriate multiset approximation spaces. It is enough to utilize only non-cooperative rules in sequential execution mode in the newly constructed P systems. To sum up, this in turn means that P systems with approximation space operating only with non-cooperative rules and in sequential execution mode can alone compensate for the features of maximally parallel rule execution, membrane dissolution and promoter/inhibitor rules.

Acknowledgements The author would like to thank the anonymous reviewers for contributing to the improvement of the paper with their valuable comments and suggestions.

Funding Open access funding provided by University of Debrecen. Research supported in part by the construction EFOP-3.6.3-VEKOP-16-2017-00002 supported by the European Union, co-financed by the European Social Fund.

References

Alhazov A, Freund R (2015) Polarizationless p systems with one active membrane. In: Rozenberg G, Salomaa A, Sempere JM, Zandron C (eds) Membrane computing. Springer, Cham, pp 51–62

Alhazov A, Aman B, Freund R (2014) P systems with anti-matter. In: Gheorghe M, Rozenberg G, Salomaa A, Sosík P, Zandron C (eds) Membrane computing. Springer, Cham, pp 66–85

Battyányi P, Vaszil G (2019) Membrane systems and multiset approximation: the cases of inner and boundary rule application. In: Mihálydeák T, Min F, Wang G, Banerjee M, Düntsch I, Suraj Z, Ciucci D (eds) Rough sets—international joint conference, IJCRS 2019, Debrecen, Hungary. Lecture notes in computer science, vol 11499. Springer, pp 239–252

Battyányi P, Mihálydeák T, Vaszil G (2019) Generalized membrane systems with dynamical structure, petri nets, and multiset approximation spaces. In: McQuillan I, Seki S (eds) Unconventional computation and natural computation, UCNC 2019, Tokyo, Japan. Lecture notes in computer science, vol 11493. Springer, pp 15–29

Csuhaj-Varjú E, Gheorghe M, Stannett M (2012) P systems controlled by general topologies. In: Durand-Lose J, Jonoska N (eds) Unconventional computation and natural computation. Springer, Berlin, pp 70–81

Csuhaj-Varjú E, Gheorghe M, Stannett M, Vaszil G (2015) Spatially localised membrane systems. Fundam Inform 138(1–2):193–205. https://doi.org/10.3233/FI-2015-1207

Freund R, Verlan S (2007) A formal framework for static (tissue) P systems. In: Eleftherakis G, Kefalas P, Păun G, Rozenberg G, Salomaa A (eds) Membrane computing. Springer, Berlin, pp 271–284

Mihálydeák T, Csajbók Z (2013) Membranes with boundaries. In: Csuhaj-Varjú E, Gheorghe M, Rozenberg G, Salomaa A, Vaszil G (eds) Membrane computing. Springer, Berlin, pp 277–294

Mihálydeák T, Csajbók Z (2014) On the membrane computations in the presence of membrane boundaries. J Autom Lang Comb 19(1):227–238

Mihálydeák T, Vaszil G (2015) Regulating rule application with membrane boundaries in P systems. In: Rozenberg G, Salomaa A, Sempere JM, Zandron C (eds) Membrane computing. Springer, Cham, pp 304–320

Păun G (2000) Computing with membranes. J Comput Syst Sci 61(1):108–143. https://doi.org/10.1006/jcss.1999.1693

Păun G (2002) Membrane computing: an introduction. Springer, Berlin

Păun G, Rozenberg G, Salomaa A (2010) The Oxford handbook of membrane computing. Oxford University Press Inc, New York

Pawlak Z (1982) Rough sets. Int J Comput Inf Sci 11(5):341–356

Pawlak Z (1991) Rough sets: theoretical aspects of reasoning about data. Kluwer Academic Publishers, Dordrecht