Flattening in (Tissue) P Systems

Rudolf Freund[1], Alberto Leporati[2], Giancarlo Mauri[2]
Antonio E. Porreca[2], Sergey Verlan[3], and Claudio Zandron[2]

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

[2] Dipartimento di Informatica, Sistemistica e Comunicazione
Università degli Studi di Milano-Bicocca
Viale Sarca 336/14, 20126 Milano, Italy
Email: alberto.leporati@unimib.it, giancarlo.mauri@unimib.it,
porreca@unimib.it, claudio.zandron@unimib.it

[3] LACL, Département Informatique, Université Paris Est
61, av. Général de Gaulle, 94010 Créteil, France
Email: verlan@univ-paris12.fr

Abstract. For many models of P systems and tissue P systems, the main behavior of a specific system can be simulated by a corresponding system with only one membrane or cell, respectively; this effective construction is called flattening. In this paper we describe the main procedure of flattening for specific variants of static (tissue) P systems as well as for classes of dynamic (tissue) P systems with a bounded number of possible membrane structures or a bounded number of cells during any computation.

1 Introduction

One of the main ideas of membrane systems as introduced by Gheorghe Păun in [8] is the distributed way of computation in the different membrane regions of a membrane system. On the other hand, even for the original variant of membrane systems using catalysts it has been shown that all computations can be carried out in only one single membrane for getting computational completeness (see [4]). Using the idea of flattening which we are going to discuss in this paper, i.e., constructing a (tissue) P system with only one membrane (cell) simulating the computations of a given membrane (tissue) P system, especially for P systems working in the sequential transition mode one often can show that the number of membranes does not matter. For example, as is well known, with transition P systems using only non-cooperative rules one can characterize the family of Parikh sets of regular languages, no matter how many membranes are used.

Whereas without any doubt for communication P systems, where computations are carried out by moving objects through membranes, the underlying membrane structure of a P system or the underlying graph structure of a tissue P system will always play an essential role, in the case of transition P systems or tissue P systems with evolution rules, a flattening procedure may allow for reducing the number of membranes or cells to one, i.e., to pure multiset rewriting, without changing the main concept for the computational power of such systems. Yet depending on the exact definitions of how these systems are supposed to use their rules and how to get the final results, specific issues have to be discussed carefully.

As this paper addresses to experts in the area of P systems, in general we only refer the reader to [9] and the P page [11] for specific notions and results used or stated afterwards. Formal definitions for a general model of static (tissue) P systems can be found in [6], a formal framework for dynamically evolving structures in [5].

2 Definitions

The set of non-negative integers is denoted by $\mathbb{N}$, the set of d-dimensional vectors of non-negative integers by $\mathbb{N}^d$. An *alphabet* V is a finite non-empty set of abstract *symbols*. Given V, the free monoid generated by V under the operation of concatenation is denoted by V^*; the elements of V^* are called strings, and the *empty string* is denoted by λ; $V^* \setminus \{\lambda\}$ is denoted by V^+. Let $\{a_1, \cdots, a_d\}$ be an arbitrary alphabet; the number of occurrences of a symbol a_i in a string x is denoted by $|x|_{a_i}$; the *Parikh vector* associated with x with respect to $a_1, \cdots, a_d$ is $\left(|x|_{a_1}, \cdots, |x|_{a_d}\right) \in \mathbb{N}^d$. The *Parikh image* of a language L over $\{a_1, \cdots, a_d\}$ is the set of all Parikh vectors of strings in L, and we denote it by $Ps\,(L)$. For a family of languages FL, the family of Parikh images of languages in FL is denoted by $PsFL$; for families of languages of a one-letter alphabet, the corresponding sets of non-negative integers are denoted by NFL. Moreover, by N^dFL we denote the family of Parikh images of languages over an alphabet of d letters in FL. Finally, we also use the convention that two sets of d-dimensional vectors in $\mathbb{N}^d$ are considered to be equal if they only differ at most by the zero-vector $(0, \cdots, 0)$.

A (finite) multiset over the (finite) alphabet V, $V = \{a_1, \cdots, a_d\}$, is a mapping $f : V \longrightarrow \mathbb{N}$ and represented by $\langle f\,(a_1)\,, a_1\rangle \cdots \langle f\,(a_d)\,, a_d\rangle$ or by any string x the Parikh vector of which with respect to $a_1, \cdots, a_d$ is $(f\,(a_1)\,, \cdots, f\,(a_d))$. In the following we will not distinguish between a vector $(m_1, \cdots, m_d)$, its representation by a multiset $\langle a_1, m_1\rangle \cdots \langle a_d, m_d\rangle$ or its representation by a string x having the Parikh vector $\left(|x|_{a_1}, \cdots, |x|_{a_d}\right) = (m_1, \cdots, m_d)$. Fixing the sequence of symbols $a_1, \cdots, a_d$ in the alphabet V in advance, the representation of the multiset $\langle m_1, a_1\rangle \cdots \langle m_d, a_d\rangle$ by the string $a_1^{m_1} \cdots a_d^{m_d}$ is unique. The set of all finite multisets over an alphabet V is denoted by $\langle V, \mathbb{N}\rangle$. If we allow some objects to appear in an unbounded number, then we consider a (finite or infinite) multiset as a mapping $f : V \longrightarrow \mathbb{N}_\infty$, where $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$ with ∞ denoting infinity. The set of all (finite or infinite) multisets over an alphabet V is denoted by $\langle V, \mathbb{N}_\infty\rangle$.

The family of regular and recursively enumerable string languages is denoted by REG and RE, respectively. For more details of formal language theory the reader is referred to the monographs and handbooks in this area as [3] and [10].

As a formal model we consider a *tissue P system of degree* $n \geq 1$ as a construct

$$\Pi = (V, T, Inf_0, \dots, Inf_n, w_0, \dots, w_n, R, f)$$

where

- V is a finite alphabet;
- $T \subseteq V$ is the terminal alphabet;
- $Inf_i \subseteq V$, for all $0 \leq i \leq n$, specifies the set of objects appearing in an unbounded number in cell i (the $n+1$ cells are labeled by $0 \dots n$ or, in a more general way, uniquely labeled by labels from a set Lab);
- $w_i \in \langle V, \mathbb{N} \rangle$, for all $0 \leq i \leq n$, is the finite multiset from $V \diagdown Inf_i$ initially associated to cell i; in total, the initial configuration of Π is described by the multisets $w_i \cup \{\langle a, \infty \rangle \mid a \in Inf_i\}$, $0 \leq i \leq n$;
- R is a finite set of *rules* of the form $(X \rightarrow Y; E)$;
- f, $0 \leq f \leq n$, is the cell where the output is collected in the generating case and the input is put in in the accepting case.

In a rule $(X \rightarrow Y; E)$, X and Y are $(n+1)$-vectors of multisets over V, i.e., $X = (x_0, \dots, x_n)$, $Y = (y_0, \dots, y_n)$, $x_i, y_i \in \langle V, \mathbb{N}_\infty \rangle$, $0 \leq i \leq n$, and E, in the most general form, is a decidable condition on the contents of the $n+1$ cells; for example, we may take $E = (P, Q)$, where $P = (p_0, \dots, p_n)$ and $Q = (q_0, \dots, q_n)$, $p_i, q_i \in \langle V, \mathbb{N} \rangle$ or finite subsets from $2^{\langle V, \mathbb{N} \rangle}$, $0 \leq i \leq n$, are permitting and forbidden contexts (for details see [6]). The application of such a rule means replacing the multiset x_i in cell i by the multiset y_i, $0 \leq i \leq n$, provided E is fulfilled; for example, for $E = (P, Q)$ this means that (every multiset from) p_i is contained in cell i whereas (any multiset from) q_i is not, for $1 \leq i \leq n$.

Transitions in a tissue P system may be carried out in the *sequential mode* (exactly one rule is applied), in the *maximally parallel mode* (an applicable multiset which cannot be extended to an applicable multiset of rules is applied), etc.; usually, a computation ends when no rule can be applied any more, i.e., *Π halts*, but there are also other ways of halting (again see [6]), e.g., stopping when a specific symbol appears. During a computation, the *configurations* of the tissue P system Π describe the finite multisets of objects from $V \diagdown Inf_i$ contained in each cell i, $0 \leq i \leq n$.

A tissue P system may be used to *generate* a (vector of) non-negative integers in a specific output cell (membrane) or to *accept* a (vector of) non-negative integers placed in a specific input cell at the beginning of a computation. Moreover, the goal can also be to *compute* an output from a given input or to output **yes** or **no** to *decide* a specific property of a given input.

If $x_i = y_i = \lambda$ for some i in a rule $(X \rightarrow Y; E)$, $X = (x_0, \dots, x_n)$, $Y = (y_0, \dots, y_n)$, $x_i, y_i \in \langle V, \mathbb{N}_\infty \rangle$, $0 \leq i \leq n$, only the remaining cells j contribute to the communication graph of Π, i.e., for each pair $(k,m) \in \{j \mid x_i \neq \lambda \text{ or } y_i \neq \lambda\}$ we introduce an (undirected) edge between the nodes k and m; if this communiction graph built up from all rules in R is a tree whose root has only one successor, then Π is called a (hierarchical) P system, with the root corresponding to the environment (usually then labeled by 0) and its single successor being the skin membrane (usuall labeled by 1), and the cells are called membranes. Usually, at least some objects occur infinitely often in the environment; these need not be taken into account within the rules with respect to the environment.

Such a hierarchical P system is represented as a construct

$$\Pi = (V, T, Env, \mu, w_0, w_1, \ldots, w_n, R, f)$$

where

- V is a finite alphabet;
- $T \subseteq V$ is the terminal alphabet;
- $Env \subseteq V$ is the set of elements appearing infinitely often in the environment (in all other membranes, all objects only appear in a finite number of copies), i.e., this corresponds to $Inf_0 = Env$ and $Inf_i = \emptyset$ for $1 \leq i \leq n$;
- μ describes the hierarchical membrane structure where 0 denotes the environment and 1 is the skin membrane;
- $w_i \in \langle V, \mathbb{N} \rangle$, for all $1 \leq i \leq n$, is the multiset initially associated to membrane i, w_0 specifies the finite multiset of objects from $V \diagdown Env$ initially appearing in the environment;
- R is a finite set of *rules* of the form $(i : X \to Y; E)$ where $1 \leq i \leq n$, $X \in \langle V, \mathbb{N} \rangle$, $Y \in 2^{(\langle V, \mathbb{N} \rangle \times TAR)}$, and

  $$TAR_\mu = \{here, in, out\} \cup \{in_j \mid 2 \leq j \leq n\}$$

  is the set of targets; the target *here* means that the generated objects remain in membrane i, $1 \leq i \leq n$; the target *out* means that the generated objects are sent out to the membrane surrounding membrane i (the parent of i, $i \geq 2$, in the graph representing the membrane structure μ); the target *in* means that the generated objects are sent into one of the membranes directly inside membrane i (one of the children of i in the graph representing the membrane structure μ), and by in_j one can directly specify one of the inner membranes where the objects are sent to;
- f, $0 \leq f \leq n$, is the membrane where the output is collected in the generating case and the input is put in in the accepting case.

Example 1. Consider the tissue P system with three cells

$$\Pi = (\{a\}, \{a\}, \emptyset, \emptyset, \emptyset, \emptyset, \langle a, 0 \rangle, \langle a, 0 \rangle, \langle a, 0 \rangle, \langle a, 0 \rangle, R, 1)$$

with

$$\begin{aligned} R = & \{(\lambda, aa, \lambda, \lambda) \to (\lambda, a, \lambda, \lambda), (\lambda, a, \lambda, \lambda) \to (\lambda, \lambda, a, \lambda)\} \\ & \cup \{(\lambda, \lambda, aa, \lambda) \to (\lambda, \lambda, \lambda, aa), (\lambda, \lambda, \lambda, aa) \to (\lambda, \lambda, aa, \lambda)\}. \end{aligned}$$

As we can see, cell 0 is not involved in any rule (so we can take it as the environment), and transitions only take place between cells 1, 2, and 3; hence, we can represent this tissue P system in a more readable way as the corresponding hierarchical P system

$$\Pi'' = (\{a\}, \{a\}, \emptyset, [\,_1 [\,_2\ [_3]\ _3\,]\ _2]\ _1, \lambda, \lambda, \lambda, \lambda, R'', 2) .$$

with

$$\begin{aligned} R'' = &\{1 : aa \rightarrow (a, here), 1 : a \rightarrow (a, in)\} \\ &\cup \{2 : aa \rightarrow (aa, in), 3 : aa \rightarrow (aa, out)\} . \end{aligned}$$

If we consider (Π and) Π'' as accepting P systems working in the maximally parallel mode, then the accepted set (of multisets) is $\left\{a^{2^n} \mid n \geq 0\right\}$, corresponding to the non-semilinear set of natural numbers $\{2^n \mid n \geq 0\}$: the rule $1 : aa \rightarrow (a, here)$ applied in parallel to the input of objects a in membrane 1 divides their number by 2; if (and only if) the original number of input objects is a power of 2, then the rule $1 : a \rightarrow (a, in)$ is only to be applied at the end of the computation; a single object a cannot be processed any more in membrane 2, whereas as soon as this rule is applied at least twice, the application of the rules $2 : aa \rightarrow (aa, in)$ and $3 : aa \rightarrow (aa, out)$ causes an infinite loop, hence, exactly the inputs a^{2^n}, $n \geq 0$, are accepted by halting computations.

3 The Basic Flattening Procedure for Static (Tissue) P Systems

Any element a in cell i of a tissue P system

$$\Pi = (V, T, Inf_0, \dots, Inf_n, w_0, \dots, w_n, R, f)$$

can be represented as a symbol (a, i) in a tissue P system

$$\Pi' = (V', T', Inf_1, w, R', 1)$$

with only one cell where

- $V' = \{(a, i) \mid a \in V, 0 \leq i \leq n\}$;
- $T' = \{(a, f) \mid a \in T\}$; especially for the generating case, only the terminal symbols in the output cell/membrane count;
- $Inf_1 = \{h_i(a_i) \mid a_i \in Inf_i,\ 1 \leq 0 \leq n\}$ where the h_i, $0 \leq i \leq n$, are the renaming morphisms $h_i : V \rightarrow V \times \{i\}$ with $h_i(a) = (a, i)$ for all $a \in V$;
- $w = h_0(w_0) \dots h_n(w_n)$;
- for getting the rules in R', any $(n+1)$-vector of multisets $(z_0, \dots, z_n)$ over V in the rules from R is replaced by the single multiset $h_0(z_0) \dots h_n(z_n)$. Similar replacements have to be taken into account for every condition E in a rule $(X \rightarrow Y; E) \in R$. For example, if $X = (x_0, \dots, x_n)$, $Y = (y_0, \dots, y_n)$, $E = (P, Q)$, $P = (p_0, \dots, p_n)$, $Q = (q_0, \dots, q_n)$, we take the corresponding rule
$$\begin{aligned} (h_0(x_0) \dots h_n(x_n) \rightarrow h_0(y_0) \dots h_n(y_n)\,; \\ ((h_0(p_0), \dots, h_n(p_n)), (h_0(q_0), \dots, h_n(q_n)))) \end{aligned}$$
into R'.

It is quite obvious that each computation step in Π' corresponds to a computation step in Π and vice versa, no matter which of the basic derivation modes – sequential, asynchronous, maximally parallel – we use; Π' working in the sequential mode now corresponds to a pure multiset rewriting system $G = (V', T', w, R')$ with permitting and forbidden contexts (provided Inf_1 is empty).

In this static case, the flattened (tissue) P system essentially is of the same kind as the original one, hence, we speak of *strong flattening*.

Example 2. Consider the accepting P system Π working in the maximally parallel mode from Example 1, then the corresponding flattened tissue P system (in the following, we omit Inf_1, as this set of objects being available infinitely often is empty) is

$$\begin{aligned}
\Pi' &= (V', T', w, R', 1)\,,\\
V' &= \{(a,i) \mid i \in \{1,2,3\}\}\,,\\
T' &= \{(a,1)\}\,,\\
w &= \lambda,\\
R &= \{(a,1)\,(a,1) \rightarrow (a,1)\,, (a,1) \rightarrow (a,2)\}\\
&\cup\ \{(a,2)\,(a,2) \rightarrow (a,3)\,(a,3)\,, (a,3)\,(a,3) \rightarrow (a,2)\,(a,2)\}\,.
\end{aligned}$$

The accepted set (of multisets) is $\left\{(a,1)^{2^n} \mid n \geq 0\right\}$, again corresponding to the non-semilinear set of natural numbers $\{2^n \mid n \geq 0\}$.

Even in the case of strong flattening, some specific small issues have to be taken into consideration carefully: as we have seen in Example 2, in the accepting case, the input vector has to be encoded by h_f, $0 \leq f \leq n$; in the generating case, in most cases we cannot avoid that we have to take the projection on the terminal alphabet (there are some simple exceptions, where all non-terminal objects have vanished whenever the computation halts, e.g., the semilinear sets, i.e., $NREG$, can be generated in that way).

Special care has to be taken for treating the environment: for tissue P systems, we may assume the environment to be one of the cells; for hierarchical P systems, the environment usually is considered to be an additional membrane with label 0; the necessary changes for Π' and especially R' are rather obvious, only the treatment of the symbols occurring infinitely often in the environment needs some special conventions (for details we refer to [6]).

4 Communication P Systems

The general model of a tissue P system also captures a lot of variants of (pure) communication P systems, e.g., P systems using antiport and symport rules. Hence, in principle, the basic flattening procedure can be applied to such communication P systems, too. Yet in this case, flattening means a dramatic change in the underlying philosophy of these system: whereas in pure communication P systems objects just move between the cells/membranes and are never destroyed or generated, in the corresponding flattened system we have rewriting rules doing exactly this kind of operations. Thus, with flattening we loose the main idea of the underlying concept. Hence, the flattened (tissue) P system is not of the same kind as the original one, and we can only speak of *flattening*, but not of strong flattening any more. We should like to point out that communication P systems with only one membrane as often occurring in the literature in fact correspond to tissue P systems with two cells, as the environment plays an essential role as the additional second cell; therefore, P systems using antiport and symport rules always need at least two cells to be represented with still capturing the idea of communication instead of rewriting, hence, in this case strong flattening is not possible.

On the other hand, flattening may still be a useful tool when investigating specific features of variants of special of communication P systems, e.g., see [1].

5 Flattening for (Tissue) P Systems with Active Membranes

In a more general case, we may allow the membranes (cells) to carry so-called polarizations from a finite set Pol; depending on those polarizations, the set of transition rules available for the objects in a membrane (cell) may vary. The unique label $h \in Lab$ and the current polarization p of a membrane (cell) can be put together in a pair (h, p) which can be taken as the new unique label of this membrane (cell); hence, using a rule changing the polarization from p to p' then means changing the label from (h, p) to (h, p'). The current structure of a (tissue) P system with polarizations can be described by a function $\mu : Lab \rightarrow Pol$ assigning one polarization to each membrane (cell). Now let M be the (finite!) set of all such functions; for each $\mu \in M$, we introduce a variable $V(\mu)$, which in the flattened system will be used as an object representing μ, and we denote $V(M) = \{V(\mu) \mid \mu \in M\}$. In our general model of a (tissue) P system Π, the rules in R now are of the form $(X \rightarrow Y; E; \mu \rightarrow \mu')$ with the rules $\mu \rightarrow \mu'$ including the changes of polarizations induced by the application of the rule $(X \rightarrow Y; E)$; we also assume that such a rule is only applicable if the current polarizations of the membranes (cells) are consistent with μ. Several such rules can only be applied in parallel if all of them together exactly yield the new structure μ'.

For Π working in one of the basic derivation modes, i.e., the sequential, asynchronous, maximally parallel mode, we immediately get the flattened (tissue) P system Π' by just using the basic flattening procedure using the extended rules and their applicability constraints as described above; moreover, the polarization μ_0 representing the structure of the start configuration of Π has to be assigned to the single membrane (cell) of Π' as its initial polarization. In general, the flattened system will be of another kind of (tissue) P systems as the original one, as we have used extended variants of rules and additional constraints for the applicability of a multiset of rules in the flattened system.

In the *sequential mode*, we can get even more: based on the construction of the flattened (tissue) P system $\mathit{\Pi}'$ as given above, we construct a tissue P system

$$\mathit{\Pi}'' = \left(V'', T', wV\left(\mu_0\right), R'', 1\right)$$

with the basic type of rules of the form $(X \to Y; E)$: the structure information from $\mathit{\Pi}$ is stored in an additional symbol; therefore, we take $V'' = V' \cup V\left(M\right)$ and start with the axiom $wV\left(\mu_0\right)$ with $V\left(\mu_0\right)$ representing the structure of the start configuration of $\mathit{\Pi}$. Moreover, we take

$$R'' = \left\{\left(uV\left(\mu\right) \to vV\left(\mu'\right); E\right) \mid \left(u \to v; E; \mu \to \mu'\right) \in R'\right\}.$$

The change of the structure now is included in (the application of) the rule itself; the only drawback of this construction is that the rules $uV\left(\mu\right) \to vV\left(\mu'\right)$ now are cooperative rules, while the original rules $u \to v$ might have been only non-cooperative rules. Yet in the best case, we even get strong flattening for (tissue) P systems with cooperative rules working in the sequential mode.

There are several other ideas for how to obtain a flattened (tissue) P system in the case of the maximally parallel mode using permitting and/or forbidding contexts which will also be discussed in the succeeding section together with membrane (cell) dissolution. Instead of one single symbol $V\left(\mu\right)$ describing the actual structure of the (tissue) P system we may also use a distinct symbol for describing the actual state of each membrane; again, the number of these symbols is finite, as we only have a finite number of membranes (cells) and corresponding polarizations. If we assume that at most one rule in any computation step may affect the status of each membrane (cell), we can use the corresponding cooperative rules $\left(uV\left(\mu_i\right) \to vV\left(\mu_i'\right); E\right)$ as already discussed earlier for the sequential mode, where the index i now indicates that only the status of membrane (cell) i is affected. But this construction is not yet sufficient for the parallel case now, as all the other rules have to *know* which current status the related membrane (cell) has; the easiest way to capture this obviously is to use permitting contexts, i.e., we have to replace each rule $(u \to v; E)$ applicable under the condition of structure μ_i by the new rule $(u \to v; E \wedge (\{\mu_i\}, \emptyset))$.

Example 3. Consider the hierarchical P system

$$\begin{aligned} \mathit{\Pi} &= \big(\left\{a\right\}, \left\{a\right\}, \emptyset, \left[\,_{(1,0)}\ \right]\,_{(1,0)}, \lambda, a, R, 1\big), \\ R &= \left\{(1,0) : a \to \left(aa, \mathit{here}\right), (1,0) : a \to \left(aa, \mathit{here}\right)\delta\left((1,0) \to (1,1)\right)\right\}. \end{aligned}$$

In the rule $(1,0) : a \to \left(aa, \mathit{here}\right)\delta\left((1,0) \to (1,1)\right)$, the part $\delta\left((1,0) \to (1,1)\right)$ indicates that with this rule the polarization of membrane 1 is changed from 0 to 1. With this new polarization, no rule is applicable any more. Therefore, if we consider $\mathit{\Pi}$ as generating P system working in the maximally parallel mode, then the generated set (of multisets) is $\left\{a^{2^n} \mid n \geq 1\right\}$, corresponding to the non-semilinear set of natural numbers $\{2^n \mid n \geq 1\}$: the rule $(1,0) : a \to (aa, \mathit{here})$ applied in parallel to objects a in membrane 1 duplicates their number; if (and only if) the rule $(1,0) : a \to \left(aa, \mathit{here}\right)\delta\left((1,0) \to (1,1)\right)$ is also applied once (at the end of the computation), the polarization of membrane 1 is changed from 0 to 1. With this new polarization, no rule is applicable any more, hence, the computation halts, with the result a^{2^n}, $n \geq 1$, with n being the number of computation steps.

In principle, this hierarchical system already has only one membrane (the environment plays no role in the evolution of the system), but we now want to construct a tissue P system with only one cell which does not change the polarization of its membrane: According to the construction described above, we obtain the following tissue P system:

$$\begin{aligned}\Pi' &= (\{(a,1)\,,\mu_{(1,0)},\mu_{(1,1)}\},\{(a,1)\}\,,(a,1)\,\mu_{(1,0)},R',1),\\ R' &= \{((a,1) \rightarrow (a,1)\,(a,1)\,;(\{\mu_{(1,0)}\},\emptyset)),\\ &\qquad ((a,1)\,\mu_{(1,0)} \rightarrow (a,1)\,(a,1)\,\mu_{(1,1)};(\{\mu_{(1,0)}\},\emptyset))\}.\end{aligned}$$

The objects $\mu_{(1,0)}$ and $\mu_{(1,1)}$ represent the polarizations 0 and 1 of menbrane 1. Starting with the initial configuration $((a,1)\,\mu_{(1,0)})$, the rule $((a,1) \rightarrow (a,1)\,(a,1)\,;(\{\mu_{(1,0)}\},\emptyset))$ can only be applied in a maximally parallel way as long as the second rule $((a,1)\,\mu_{(1,0)} \rightarrow (a,1)\,(a,1)\,\mu_{(1,1)};(\{\mu_{(1,0)}\},\emptyset))$ is not applied, too, whereafter the computation immediately halts.

6 Flattening for (Tissue) P Systems with Membrane (Cell) Dissolution

Already in the original model of membrane systems introduced in [8], the possibility of membrane dissolution was investigated. The objects from the dissolved membrane r are moved into the surrounding membrane region. In a more general context, the dissolution of a cell and the moving of its contents were discussed in [5] as the operation *Delete–and–Move*(r). Such systems with membrane (cell) dissolution have a finite number of possible membrane structures, as the dissolution operation can only decrease the number of membranes (cells) already present at the beginning of the computation. Hence, it is possible to mimic the effect of the dissolution by assigning a marker to each membrane in order to indicate if the membrane is dissolved or not, and by using permitting and/or forbidding contexts in order to check these markers and by using a subset construction at the level of rules in order to capture all possible structure changes.

The main idea for the flattening procedure is that the objects from a deleted membrane (cell) i are moved to another membrane (cell) j and there are treated as objects from membrane (cell) j, i.e., every object (a,i) now has to be treated as an object (a,j). Hence, for any possible membrane (cell) structure μ we define a mapping I_μ interpreting the objects (a,i) with respect to the current membrane (cell) structure μ. In the flattened (tissue) P system even with polarizations as described before we then have the rules $(I_\mu\,(u) \rightarrow I_{\mu'}\,(v)\,;I_\mu\,(E)\,;\mu \rightarrow \mu')$ instead of the rules $(u \rightarrow v;E;\mu \rightarrow \mu')$ – obviously the condition E has to be interpreted in the sense of I_μ, too. For technical details concerning the formal interpretation of the structure changes caused by $\mu \rightarrow \mu'$ including deletion of a membrane (cell) with moving its contents to the surrounding membrane region (to another cell) we refer the expert reader to [5].

For the sequential mode, according to the construction given in the preceding section, we get the tissue P system

$$\Pi'' = \left(V'', T', I_{\mu_0}(w) V(\mu_0), R'', 1\right)$$

with

$$R'' = \{(I_\mu(u) V(\mu) \to I_{\mu'}(v) V(\mu'); I_\mu(E)) \mid (u \to v; E; \mu \to \mu') \in R'\}.$$

For hierarchical P systems working in the maximally parallel derivation mode, a flattening procedure was described in [2]. The main idea of such a proof is that, besides taking the additional rules $V(\mu) \to V(\mu')$ for all possible membrane structures μ, μ', the maximally parallel application of the original rules together with exactly one of these rules is controlled by taking $V(\mu)$ as (eventually additional) permitting context in a similar way as we have already discussed in the preceding section, e.g., see Example 3. With such a construction, even the dissolution of several membranes in one computation step can be captured.

In the following example, we now follow the idea already exhibited at the beginning of this section with describing the status (existing/not existing) of each membrane (cell) by a distinct symbol; moreover, each object in a rule, according to these symbols (characterizing the actual structure of the system) given as permitting contexts, may originate from different membranes (cells).

Example 4. Consider the hierarchical P system

$$\begin{aligned} \Pi &= (\{a,b\}, \{b\}, \emptyset, [\,_1 [\,_2\,]\,_2 [\,_3\,]\,_3\,]\,_1, \lambda, \lambda, a, a, R, 1) \\ R &= \left\{2 : a \to \left(a^2, here\right), 2 : a \to \left(a^2, here\right)\delta\right\} \\ &\cup \left\{3 : a \to \left(a^4, here\right), 3 : a \to \left(a^4, here\right)\delta\right\} \\ &\cup \left\{1 : aa \to \left(b^3, here\right)\right\}. \end{aligned}$$

We consider Π as generating P system working in the maximally parallel mode: in membranes 2 and 3, in each computation step, the actual number of objects is multiplied by 2 and 4 by the rules $2 : a \to \left(a^2, here\right)$ and $3 : a \to \left(a^4, here\right)$, respectively. By applying the rules with the membrane dissolution operator δ in one of these membranes, the corresponding membrane is dissolved and the objects a are sent to the skin membrane, where in the next step, from each couple of objects a three terminal objects b evolve. As soon as both membranes 2 and 3 have been dissolved, the system halts. In sum, the generated set (of multisets) corresponds to the non-semilinear set of natural numbers $\left\{3.2^{n-1} \mid n \geq 1\right\} \cup \left\{6.4^{m-1} \mid m \geq 1\right\}$ where n and m are the numbers of computation steps in membranes 2 and 3 until the dissolution of the corresponding membrane.

As the environment is not involved in the P system Π, the corresponding flattened tissue P system can be constructed as follows:

$$\begin{aligned}
\Pi' &= (V', T', w, R', 1)\,,\\
V' &= \{(a,i)\,,(b,i) \mid i \in \{1,2,3\}\} \cup \{s_i, \bar{s}_i \mid i \in \{1,2,3\}\}\,,\\
T' &= \{(b,1)\}\,,\\
w &= (a,2)\,(a,3)\,,\\
R' &= \{((a,2) \to (a,2)^2\,; (\{s_2\}\,,\emptyset)), ((a,2)\,s_2 \to (a,2)^2\,\bar{s}_2; (\{s_2\}\,,\emptyset))\}\\
&\cup\ \{((a,3) \to (a,3)^4\,; (\{s_3\}\,,\emptyset)), ((a,3)\,s_3 \to (a,3)^4\,\bar{s}_3; (\{s_3\}\,,\emptyset))\}\\
&\cup\ \{(a,1)\,(a,1) \to (b,1)^3\,; (\{s_1\}\,,\emptyset)), (a,1)\,(a,2) \to (b,1)^3\,; (\{s_1, \bar{s}_2\}\,,\emptyset)),\\
&\qquad (a,2)\,(a,2) \to (b,1)^3\,; (\{s_1, \bar{s}_2\}\,,\emptyset)), (a,1)\,(a,3) \to (b,1)^3\,; (\{s_1, \bar{s}_3\}\,,\emptyset)),\\
&\qquad (a,3)\,(a,3) \to (b,1)^3\,; (\{s_1, \bar{s}_3\}\,,\emptyset)),\\
&\qquad (a,2)\,(a,3) \to (b,1)^3\,; (\{s_1, \bar{s}_2, \bar{s}_3\}\,,\emptyset))\}.
\end{aligned}$$

The symbols s_i ($\bar{s}_i$) indicate that membrane i is (not) present. In membrane 1, the symbol a may originate from membranes 1, 2, and 3; hence, for the left-hand side of the original rule $1 : aa \to \left(b^3, here\right)$ each of the two symbols a may come from each of the three membranes depending on the underlying membrane structure which is visible from the permitting context. As the skin membrane (membrane 1) must not be deleted, each occurrence of s_1 in the permitting contexts can be omitted.

When the computation halts, the symbols $s_1, \bar{s}_2, \bar{s}_3$ are present. In order to eliminate these additional symbols (instead of having to make the projection on the terminal alphabet $\{(b,1)\}$), we would have to add the rule

$$(s_1\bar{s}_2\bar{s}_3 \to \lambda; (\emptyset, \{(a,1)\,,(a,2)\,,(a,3)\}))$$

with the forbidden context guaranteeing that no rule can be applied any more. This idea with such a forbidden context can be used in general, too.

7 Flattening for (Tissue) P Systems with Membrane (Cell) Creation, Division, and Dissolution

Whereas the deletion of membranes (cells) still allows for flattening using specific constructions, as soon as membrane (cell) division and/or creation are allowed, in general the number of membranes (cells) need not be bounded any more. Hence, a naive adaptation of the flattening procedure as described above would lead to potentially infinite numbers of objects and rules. On the other hand, if in any computation of the system the number of possible structures (membranes/cells) can be bounded by some constant max, then similar constructions as given for systems with membrane (cell) dissolution may yield a flattened system.

Example 5. Consider the P system (with active membranes)

$$\begin{aligned}
\Pi &= (\{a,b\}\,,\{a\}\,,\emptyset, [\,_1]\,_1, \lambda, b, R, 1)\\
R &= \{1 : b \to [\,_2\,a]\,_2\,[\,_2\,a]\,_2\}\\
&\cup\ \left\{2 : a \to \left(a^2, here\right), 2 : a \to \left(a^2, here\right)\delta\right\}.
\end{aligned}$$

We consider Π as a generating P system working in the maximally parallel mode: starting with the initial configuration $[\,_1 b]\,_1$, the rule $1 : b \to [\,_2\, a]\,_2\, [\,_2\, a]\,_2$ creates two inner membranes with the same label 2, i.e., we obtain the configuration $[\,_1\, [\,_2\, a]\,_2\, [\,_2\, a]\,_2\,]\,_1$. In both membranes, in each computation step, the actual number of objects a is duplicated by the rules $2 : a \to \left(a^2, here\right)$ until the application of at least one rule $2 : a \to \left(a^2, here\right)\delta$ dissolves the membrane. As in membrane 1 no rule can be applied to objects a, at the end, we obtain k objects a with $k \in \{2^m + 2^n \mid m, n \geq 1\}$.

A specific problem of such P systems with active membranes is that we may generate new membranes having the same label. Yet the assumption that only a finite number of different membrane structures may arise allows us to assign different labels to copies of membranes having the same label, e.g., for the two membranes with label 2 in Π we may take the labels $(2,1)$ and $(2,2)$; in that way we obtain an equivalent P system Π'' where we have to duplicate the rules for membrane 2 for the two membranes now labeled by $(2,1)$ and $(2,2)$:

$$\begin{aligned}
\Pi'' &= (\{a,b\}, \{a\}, \emptyset, [\,_1]\,_1, \lambda, b, R'', 1)\\
R'' &= \left\{1 : b \to \left[\,_{(2,1)}\, a\right]\,_{(2,1)} \left[\,_{(2,2)}\, a\right]\,_{(2,2)}\right\}\\
&\quad \cup \left\{(2,1) : a \to \left(a^2, here\right), (2,1) : a \to \left(a^2, here\right)\delta\right\}\\
&\quad \cup \left\{(2,2) : a \to \left(a^2, here\right), (2,2) : a \to \left(a^2, here\right)\delta\right\}.
\end{aligned}$$

As the environment 0 is not involved in the P systems Π and Π'', for Π'' an equivalent flattened tissue P system can be constructed as follows using extended rules of the form $(I_\mu(u) \to I_{\mu'}(v); \mu \to \mu')$, i.e., the left-hand sides of the rules are interpreted according to the current membrane structure μ, whereas the evolving objects from the right-hand sides have to be interpreted already as objects in the new membrane structure μ':

$$\begin{aligned}
\Pi' &= (V', T', w, R', 1),\\
V' &= \{(a,i), (b,i) \mid i \in \{1, (2,1), (2,2)\}\},\\
T' &= \{(a,1)\},\\
w &= (b,1),\\
R' &= \{\big((b,1) \to (a,(2,1))\,(a,(2,2)); [\,_1]\,_1 \to \left[\,_1 \left[\,_{(2,1)}\right]\,_{(2,1)} \left[\,_{(2,2)}\right]\,_{(2,2)}\right]\,_1\big)\}\\
&\quad \cup \{((a,(2,1)) \to (a,(2,1))^2; \left[\,_1 \left[\,_{(2,1)}\right]\,_{(2,1)} \left[\,_{(2,2)}\right]\,_{(2,2)}\right]\,_1 \to \mu'),\\
&\qquad \mu' \in \left\{\left[\,_1 \left[\,_{(2,1)}\right]\,_{(2,1)} \left[\,_{(2,2)}\right]\,_{(2,2)}\right]\,_1, \left[\,_1 \left[\,_{(2,1)}\right]\,_{(2,1)}\right]\,_1\right\}\}\\
&\quad \cup \{((a,(2,2)) \to (a,(2,2))^2; \left[\,_1 \left[\,_{(2,1)}\right]\,_{(2,1)} \left[\,_{(2,2)}\right]\,_{(2,2)}\right]\,_1 \to \mu'),\\
&\qquad \mu' \in \left\{\left[\,_1 \left[\,_{(2,1)}\right]\,_{(2,1)} \left[\,_{(2,2)}\right]\,_{(2,2)}\right]\,_1, \left[\,_1 \left[\,_{(2,2)}\right]\,_{(2,2)}\right]\,_1\right\}\}\\
&\quad \cup \{((a,(2,1)) \to (a,1)^2; \left[\,_1 \left[\,_{(2,1)}\right]\,_{(2,1)} \left[\,_{(2,2)}\right]\,_{(2,2)}\right]\,_1 \to \mu'),\\
&\qquad \mu' \in \left\{\left[\,_1 \left[\,_{(2,2)}\right]\,_{(2,2)}\right]\,_1, [\,_1]\,_1\right\}\}\\
&\quad \cup \{((a,(2,2)) \to (a,1)^2; \left[\,_1 \left[\,_{(2,1)}\right]\,_{(2,1)} \left[\,_{(2,2)}\right]\,_{(2,2)}\right]\,_1 \to \mu'),\\
&\qquad \mu' \in \left\{\left[\,_1 \left[\,_{(2,1)}\right]\,_{(2,1)}\right]\,_1, [\,_1]\,_1\right\}\}.
\end{aligned}$$

The condition that only rules yielding the same μ' can be applied in parallel guarantees that the correct symbols are evolving with respect to the next membrane structure μ'. Again, for any halting computation, the final configuration is of the form $\left[\,_1 a^k\right]\,_1$ with $k \in \{2^m + 2^n \mid m, n \geq 1\}$.

We have to point out that a construction like that given above only works with the maximally parallel derivation mode, as with the change of the structure all symbols from a dissolved membrane have to be converted into symbols of the membrane they are sent to by the dissolution of the membrane. On the other hand, in the preceding example, after the first derivation step, each of the possible membrane structures evolves by dissolution only, hence, we could also apply each of the flattening techniques as described in the preceding section.

In the following example, the membrane structure may evolve from $[_1\,]_1$ to $[_1[_2\,]_2\,]_1$ and back an unbounded number of times, hence, the parallel rewriting of all symbols is a crucial point of the flattening procedure:

Example 6. Consider the P system (with active membranes)

$$\begin{aligned}
\Pi &= (\{a,b,c\}, \{c\}, \emptyset, [_1\,]_1, \lambda, ba, R, 1)\\
R &= \left\{1 : b \to [_2\, c]_2, 1 : b \to \lambda, 1 : a \to \left(c^3, here\right), 1 : c \to \left(a^2, in\right)\right\}\\
&\quad \cup \{2 : c \to (b, here), 2 : b \to (b, out)\,\delta\}.
\end{aligned}$$

We consider Π as a generating P system working in the maximally parallel mode: starting with the initial configuration $[_1 ba]_1$, the rule $1 : b \to [_2\, c]_2$ creates an inner membrane with label 2, whereas by the rule $1 : a \to \left(c^3, here\right)$ from each object a we get three symbols c, i.e., after the first derivation step we obtain the configuration $\left[_1 c^3 [_2\, c]_2\right]_1$. In the next computation step, from each object c we get two objects a which are sent into membrane 2 by the rules $1 : c \to \left(a^2, in\right)$, and at the same time, in membrane 2 the single object c evolves back to b by the rule $2 : c \to (b, here)$, i.e., after this derivation step we have got the configuration $\left[_1 \left[_2\, a^6 b\right]_2\right]_1$. With the dissolution of membrane 2 – using the rule $2 : b \to (b, out)\,\delta$ – all objects a and the single object b are back in membrane 1. This cycle continues as long as in membrane 1 the rule $1 : b \to [_2\, c]_2$ is applied, whereas the derivation halts after the application of the rule $1 : b \to \lambda$. In that way, Π generates $\left\{c^{3.6^n} \mid n \geq 0\right\}$.

As the environment 0 is not involved in the P systems Π, for Π an equivalent flattened tissue P system can be constructed as follows using extended rules of the form $(I_\mu(u) \to I_{\mu'}(v); \mu \to \mu')$:

$$\begin{aligned}
\Pi' &= (V', T', w, R', 1),\\
V' &= \{(a,i), (b,i), (c,i) \mid i \in \{1,2\}\},\\
T' &= \{(c,1)\},\\
w &= (ab,1),\\
R' &= \{((b,1) \to (c,2); [_1\,]_1 \to [_1[_2\,]_2]_1),\\
&\qquad ((b,1) \to \lambda; [_1]_1 \to [_1]_1),\\
&\qquad ((a,1) \to (c,1)^3; [_1]_1 \to [_1[_2\,]_2]_1),\\
&\qquad ((c,1) \to (a,2)^2; [_1[_2\,]_2]_1 \to [_1[_2\,]_2]_1)\}\\
&\quad \cup \{((c,2) \to (b,2); [_1[_2\,]_2]_1 \to [_1[_2\,]_2]_1),\\
&\qquad ((b,2) \to (b,1); [_1[_2\,]_2]_1 \to [_1]_1)\}\\
&\quad \cup \{((a,2) \to (a,1); [_1[_2\,]_2]_1 \to [_1]_1)\}.
\end{aligned}$$

As they never would be applicable, rules like $((a,1) \to (c,1)^3 ; [_1]_1 \to [_1]_1)$ and $((c,2) \to (b,1) ; [_1 [_2]_2]_1 \to [_1]_1)$ were omitted, whereas the rule $((a,2) \to (a,1) ; [_1 [_2]_2]_1 \to [_1]_1)$ had to be added to rename the symbols a appearing as objects $(a,2)$ in membrane 1 into objects $(a,1)$ when membrane 2 is dissolved. For any halting computation, the final configuration is of the form $\left[_1 (c,1)^{3.6^n} \right]_1$ for some $n \geq 0$, i.e., as Π also Π' generates the non-semilinear set $\{3.6^n \mid n \geq 0\}$.

Finally, we consider a simple hierarchical P system with non-elementary membrane division, where the contents of the original cell is duplicated into two new cells:

Example 7. Consider the P system (with active membranes)

$$\begin{aligned} \Pi &= (\{a,b\}, \{a\}, \emptyset, [_1]_1, \lambda, b, R, 1) \\ R &= \{0 : [_1 b]_1 \to [_2 b]_2 [_3 b]_3, 2 : b \to \delta, 3 : b \to \delta\}. \end{aligned}$$

We consider Π as a computing P system working in the maximally parallel mode, with the input a^n, $n \geq 0$, being given in membrane 1, and the output collected in the environment (membrane 0): starting with the initial configuration $[_1 ba^n]_1$, the rule $[_1 b]_1 \to [_2 b]_2 [_3 b]_3$ divides membrane 1 and copies its contents into two new membranes 2 and 3; in the second step, the single objects b in membranes 2 and 3 dissolve the membranes, thus releasing their contents to the environment. Hence, computation stops after two steps with a^{2n} in the environment as the result of the computation.

An equivalent flattened tissue P system Π' can be constructed as follows using extended rules of the form $(I_\mu (u) \to I_{\mu'} (v) ; \mu \to \mu')$; again, we only include those rules which can be applied during a computation:

$$\begin{aligned} \Pi' &= (V', T', w, R', 1), \\ V' &= \{(a,i), (b,i) \mid i \in \{0,1,2,3\}\}, \\ T' &= \{(a,0)\}, \\ w &= (b,1), \\ R' &= \{((b,1) \to (b,2)(b,3) ; [_0 [_1]_1]_0 \to [_0 [_2]_2 [_3]_3]_0), \\ &\qquad ((a,1) \to (a,2)(a,3) ; [_0 [_1]_1]_0 \to [_0 [_2]_2 [_3]_3]_0)\} \\ &\quad \cup \{((b,2) \to \lambda; [_0 [_2]_2 [_3]_3]_0 \to [_0]_0), \\ &\qquad ((b,3) \to \lambda; [_0 [_2]_2 [_3]_3]_0 \to [_0]_0), \\ &\qquad ((a,2) \to (a,1) ; [_0 [_2]_2 [_3]_3]_0 \to [_0]_0), \\ &\qquad ((a,3) \to (a,1) ; [_0 [_2]_2 [_3]_3]_0 \to [_0]_0)\}. \end{aligned}$$

Π' starts with the axiom $(b,1)$ and the (additional) input $(a,1)^n$ in its single cell; by applying the first two rules we obtain $(b,2)(a,2)^n (b,3)(a,3)^n$, whereas the remaining rules are used in the second computation step to obtain the result of the computation $(a,1)^{2n}$.

8 Flattening with Changing the Transition Mode

Several models of tissue P systems work in such a way that in each cell one rule is applied (if possible), but in one computation step such a sequential derivation has to take place in all cells, i.e., such systems work sequentially on the level of the cells, but in a maximally parallel way on the level of the whole system. Examples for such models are spiking neural P systems or variants of enzymatic numerical P systems considered in several papers just recently (e.g., see [7] and the references therein).

The basic flattening procedure may be applied to the objects in such systems as usual, but in the single membrane of the flattened system Π' to these objects the original rules now have to be applied in the min_1 transition mode: the new rule set R' is the union of the original rule sets R_0 to R_n asociated with the cells of the original system, but for the application of the min_1 transition mode again divided into the sets R_0 to R_n, i.e., from each set R_i, $0 \leq i \leq n$, exactly one rule (if possible) is taken for any multiset to be applied in a computation step of Π'.

9 Final Remarks

In this paper we have discussed the flattening procedure for several of the most common models of (tissue) P systems in a general framework of (tissue) P systems, even with membrane (cell) dissolution and polarizations, but without membrane (cell) generation or division, as in this case the number of membranes (cells) in general is not bounded. For (tissue) P systems with a bounded number of cells during any computation, flattening even works for dynamically changing structures. In sum, several models of membrane systems can be reduced to pure multiset rewriting by flattening these systems to one membrane (cell), but in general a lot of interesting features arising from the idea to distribute the objects and their evolution into different membranes (cells) remains still valid.

References

1. M. Antoniotti, R. Freund, A. Leporati, G. Mauri: Self-stabilization in membrane systems. *The Computer Science Journal of Moldova* **20** (2), 133–146 (2012).
2. O. Agrigoroaiei, G. Ciobanu: Flattening the transition P systems with dissolution. CMC11, 2010, *Lecture Notes in Computer Science* **6501**, Springer, 2010, 53–64.
3. J. Dassow, Gh. Păun: *Regulated Rewriting in Formal Language Theory.* Springer, 1989.
4. R. Freund, L. Kari, M. Oswald, P. Sosík: Computationally universal P systems without priorities: two catalysts are sufficient. *Theor. Comp. Sci.* **330**, 251–266 (2005).
5. R. Freund, I. Pérez-Hurtado, A. Riscos-Núñez, S. Verlan: A formalization of membrane systems with dynamically evolving structures. *Int. J. Comput. Math.* **90** (4), 801–815 (2013).
6. R. Freund, S. Verlan: A formal framework for static (tissue) P systems. WMC8, 2007, *Lecture Notes in Computer Science* **4860**, Springer, 2007, 271–284.
7. A. Leporati, G. Mauri, A. E. Porreca, Claudio Zandron: Improved universality results for parallel enzymatic numerical P systems, *submitted to IJUC.*
8. Gh. Păun: Computing with membranes. *J. Comput. Syst. Sci.* **61**, 108–143 (2000) (see also TUCS Report 208, November 1998, `www.tucs.fi`).

9. Gh. Păun, G. Rozenberg, A. Salomaa (Eds.): *The Oxford Handbook of Membrane Computing*. Oxford University Press, 2010.
10. G. Rozenberg, A. Salomaa (Eds.): *Handbook of Formal Languages,* 3 volumes. Springer, 1997.
11. The P Systems Website: `http://ppage.psystems.eu`.