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

One of the main ideas of membrane systems as introduced by Gheorghe Păun in [7] 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 [3]). Using the idea of flattening which we are going to discuss in this note, 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.

This note addresses to experts in the area of P systems; hence, in general we only refer the reader to [8] and the P page [9] for specific notions and results used or stated afterwards. Formal definitions for a general model of static (tissue) P systems can be found in [5], a formal framework for dynamically evolving structures in [4].

As a formal model we consider a *tissue P system of degree* $n \geq 1$ as a construct $\Pi = (V, T, w_1, w_2, \ldots, w_n, R, f)$ where V is a finite alphabet; $T \subseteq V$

is the terminal alphabet; $w_i \in \langle V, \mathbb{N} \rangle$ ($\langle V, \mathbb{N} \rangle$ denotes the set of finite multisets over V), for all $1 \leq i \leq n$, is the multiset initially associated to cell i (the n cells are labeled by $1 \dots n$ or, in a more general way, uniquely labeled by labels from a set Lab); R is a finite set of *rules* of the form $(X \to Y; E)$, and f 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 $(X \to Y; E)$, X and Y are n-vectors of multisets over V, i.e., $X = (x_1, \dots, x_n)$, $Y = (y_1, \dots, y_n)$, $x_i, y_i \in \langle V, \mathbb{N} \rangle$, $1 \leq i \leq n$, and E, in the most general form, is a decidable condition on the contents of the n cells; e.g., we may take $E = (P, Q)$, where $P = (p_1, \dots, p_n)$ and $Q = (q_1, \dots, q_n)$, $p_i, q_i \in \langle V, \mathbb{N} \rangle$ or finite subsets from $2^{\langle V, \mathbb{N} \rangle}$, $1 \leq i \leq n$, are permitting and forbidden contexts (for details see [5]). The application of such a rule means replacing the multiset x_i in cell i by the multiset y_i, $1 \leq i \leq n$, provided E is fulfilled, e.g., (every multiset from) p_i is contained in cell i whereas (any multiset from) q_i is not, for $1 \leq i \leq n$.

If $x_i = y_i = \lambda$ for some i in a rule, only the remaining cells contribute to the communication graph of Π; if this communication 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 and its single successor being the skin membrane, 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.

Transitions in a tissue P system may be carried out in the *sequential mode* (exactly one rule is applied), in the *maximally parallel mode* 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 [5]), e.g., stopping when a specific symbol appears.

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.

The Basic Flattening Procedure for Static (Tissue) P Systems

Any element a in cell i of a tissue P system $\Pi = (V, T, w_1, w_2, \dots, w_n, R, f)$ can be represented as a symbol (a, i) in a tissue P system $\Pi' = (V', T', w, R', 1)$ with only one cell, i.e., $V' = \{(a, i) \mid a \in V, 1 \leq i \leq n\}$ and, especially for the generating case, $T' = \{(a, f) \mid a \in T\}$ (only the terminal symbols in the output cell/membrane count); moreover, $w = h_1(w_1) \dots h_n(w_n)$. Any n-vector of multisets $(z_1, \dots, z_n)$ over V in the rules from R is replaced by the single multiset $h_1(z_1) \dots h_n(z_n)$ where the h_i, $1 \leq i \leq n$, are the renaming morphisms $h_i : V \to V \times \{i\}$ with $h_i(a) = (a, i)$ for all $a \in V$. Similar replacements have to be taken into account for every condition E in a rule $(X \to Y; E) \in R$. For example, if $X = (x_1, \dots, x_n)$, $Y = (y_1, \dots, y_n)$, $E = (P, Q)$, $P = (p_1, \dots, p_n)$, $Q = (q_1, \dots, q_n)$, we take the corresponding rule

$$\begin{aligned} &(h_1(x_1) \dots h_n(x_n) \to h_1(y_1) \dots h_n(y_n); \\ &\quad ((h_1(p_1), \dots, h_n(p_n)), (h_1(q_1), \dots, h_n(q_n)))) \end{aligned}$$

into R'; in that sense, Π' working in the sequential mode now corresponds to a pure multiset rewriting system $G = (V', T', w, R')$ with permitting and forbidden contexts.

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. Only some small issues have to be taken into consideration carefully: in the generating case, we cannot avoid that we have to take the intersection with the terminal alphabet; in the accepting case, the input vector has to be encoded by the h_i, $1 \leq i \leq n$. 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 [5]).

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. Hence, with flattening we loose the main idea of the underlying concept. 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].

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; 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 $\mu \in M$, let $V(\mu)$ denote the variable representing μ, and $V(M) = \{V(\mu) \mid \mu \in M\}$. 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 $\mu \rightarrow \mu'$ capturing the changes of polarizations; 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 yield the same new structure μ'.

For $\mathit{\Pi}$ working in one of the basic derivation modes, i.e., the sequential, asynchronous, maximally parallel mode, we immediately get the flattened (tissue) P system $\mathit{\Pi}'$ by just using the basic flattening procedure and additionally assigning the polarization μ_0 representing the structure of the start configuration of $\mathit{\Pi}$ to the single membrane (cell) of $\mathit{\Pi}'$ as its initial polarization.

In the sequential mode, we can get even more: based on the construction of $\mathit{\Pi}'$ given above, we construct a tissue P system $\mathit{\Pi}'' = (V'', T', wV(\mu_0), R'', 1)$ 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 $V'' = V' \cup V(M)$ and start with the axiom $wV(\mu_0)$ with $V(\mu_0)$ representing the structure of the start configuration of $\mathit{\Pi}$. Moreover, we take $R'' = \{(uV(\mu) \to vV(\mu'); E) \mid (u \to v; E; \mu \to \mu') \in R'\}$. The only drawback of this construction is that the rules $uV(\mu) \to vV(\mu')$ now are cooperative rules, while the original rules $u \to v$ might have been only non-cooperative rules.

Flattening for (Tissue) P Systems with Dissolution

Already in the original model of membrane systems introduced in [7], 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 [4] as the operation *Delete–and–Move*(r). The main idea for the flattening procedure is that the objects from the deleted cell are moved to another cell r' and there are treated as objects from cell r'. Hence, let I_μ be the interpretation of the objects (a, i) with respect to the membrane (cell) structure μ. In the flattened (tissue) P system even with polarizations as described before we then have the rules $(I_\mu(u) \to I_\mu(v); I_\mu(E); \mu \to \mu')$ instead of the rules $(u \to v; E; \mu \to \mu')$ – 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 $\mu \to \mu'$ including deletion of a membrane (cell) with moving its contents to the surrounding membrane region (to another cell) we refer the expert reader to [4]. For the sequential mode, according to the preceding construction, we get the tissue P system $\mathit{\Pi}'' = (V'', T', I_{\mu_0}(w)V(\mu_0), R'', 1)$ 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; to ensure correct halting, for the rules $V(\mu) \to V(\mu')$ in addition any left-hand side of the other rules has to be taken as a possible permitting context.

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 [6] 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_1 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_1 to R_n, i.e., from each set R_i, $1 \leq i \leq n$, exactly one rule (if possible) is taken for any multiset to be applied in a computation step of Π'.

Final Remarks

In this note 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, flattening may even work for dynamically changing structures, but there may be the need to consider more complex evolution rules or even different derivation modes. Therefore, although several models of membrane systems can be reduced to pure multiset rewriting by flattening these systems to one membrane (cell), 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. A. Alhazov, 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. 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).
4. 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).
5. R. Freund, S. Verlan: A formal framework for static (tissue) P systems. WMC8, 2007, *Lecture Notes in Computer Science* **4860**, Springer, 2007, 271–284.
6. A. Leporati, G. Mauri, A. E. Porreca, Claudio Zandron: Improved universality results for parallel enzymatic numerical P systems, *submitted to IJUC*.
7. Gh. Păun: Computing with membranes. *J. Comput. Syst. Sci.* **61**, 108–143 (2000) (see also TUCS Report 208, November 1998, `www.tucs.fi`).
8. Gh. Păun, G. Rozenberg, A. Salomaa (Eds.): *The Oxford Handbook of Membrane Computing*. Oxford University Press, 2010.
9. The P Systems Website: `http://ppage.psystems.eu`.