
Towards a P Systems Normal Form Preserving Step-by-step Behavior

Roberto Barbuti¹, Andrea Maggiolo-Schettini¹, Paolo Milazzo¹, Simone Tini²

¹ Dipartimento di Informatica, Università di Pisa
Largo Pontecorvo 3, 56127 Pisa, Italy

² Dip. di Scienze della Cultura, Politiche e dell'Informazione
Università dell'Insubria
Via Valleggio 11, 22100 Como, Italy

Summary. Starting from a compositional operational semantics of transition P Systems we have previously defined, we face the problem of developing an axiomatization that is sound and complete with respect to some behavioural equivalence. To achieve this goal, we propose to transform the systems into a unique normal form which preserves the semantics. As a first step, we introduce axioms which allow the transformation of membrane structures with no dissolving rules into flat membranes. We discuss the problems which arise when dissolving rules are allowed and we suggest possible solutions. We leave as future work the further step that leads to the wanted normal form.

1 Introduction

We have recently defined a compositional operational semantics of P Systems as a labeled transition system (LTS) [2]. The class of P Systems we have considered are the so called *transition P Systems* with dissolving rules and without restrictions on evolution rules. In the definition of the semantics, P Systems are seen as *reactive systems*, namely as systems that can receive stimuli from an environment and can react to these stimuli, possibly by sending some reply back to the environment. In particular, membranes are seen as the entities that can receive stimuli, in terms of objects, from an environment. The environment of a membrane can be another membrane containing it and having some rules which send objects into it. As an environment of a membrane we consider also other membranes possibly contained in it and having some rules which send objects out. The objects received by a membrane from the environment could enable the application of some rules of the membrane that eventually could send some objects back to the environment, namely to outer and inner membranes.

The LTS we have defined allows us to observe the behavior of membranes in terms of objects sent to and received from inner and external membranes. A state of the LTS is a configuration of the considered P System, and a transition

from a state to another describes an execution step of the P System, in which rules are applied according to maximal parallelism in all the membranes of the system. Transitions are labeled with the multiset of objects received from the environment, the multiset of objects sent to outer membranes, and the multiset of objects sent to inner membranes in the described execution step. Other information carried by labels is needed to build the LTS in a compositional way. This means that the semantics of a complex system can be inferred from the semantics of its components.

In [2] we have proved that some well-known behavioural equivalences such as trace equivalence and bisimulation defined on our LTS are congruences. This means that if we can prove that a membrane system that is a component of some bigger system is behaviourally equivalent to another membrane system, then the former can be replaced with the latter in the bigger system without changing the global behaviour. In other words, there exists no environment in which the bigger system with the original component reacts differently to stimuli with respect to the same system with the replaced component.

Behavioural equivalences are powerful analysis tools as they allow us to compare the behaviours of two systems and to verify properties of a system by assessing the equivalence between such a system and another one known to satisfy those properties. However, proving behavioural equivalence is not easy because the semantics of a system often consists of infinite states and infinite transitions. For this reason, it is usually important to find an axiomatization of some behavioural equivalence, namely a sound and complete characterization of the equivalence in terms of axioms on the syntax of systems. In this way the equivalence between two systems could be proved by showing that there exists a sequence of applications of such axioms that transform one system into the other. This allows the proof of the equivalence to be performed without considering the (possibly complex) semantics of the compared systems, and this usually favors the development of tools for the comparison of systems.

We would like to define a sound and complete axiomatization of the semantics we have given in [2]. It is not easy to prove soundness and completeness of an axiomatization, namely that axioms relate behaviorally equivalent systems and that all behaviourally equivalent systems are in the relation characterized by the axioms. In particular, completeness proof is usually difficult. What can help to prove this result is a notion of normal form to which all the considered systems can be reduced. This could allow the set of axioms to be split into two subsets: one consisting of the axioms that can be used to bring the systems into their normal form and the other consisting of axioms that relate systems in normal form. This, in turn, could allow the proof of completeness to be simplified by considering only the axioms in the second set.

In this paper we perform the first steps towards the definition of a normal form for P Systems preserving behavioural equivalences. In particular, we face the problem of determining the membrane structure of the normal form of a system. We start by considering P Systems without dissolving rules, and we show, by

giving a few axioms, that any P System in this class can be transformed into an equivalent P System consisting of one only membrane (a *flat* system). In order to obtain this result we slightly enrich the membranes of a P System, namely we associate with each membrane an *interface*, that is a set of objects that are allowed to be received by the membrane from the environment.

We also discuss the problem of considering P Systems containing dissolving rules. We show that in this case it is no longer possible to find an equivalent flat form, but we discuss how an alternative “standard” form can be reached.

Related work

Operational semantics for P Systems have been proposed in [1, 5, 6, 8]. All these semantics are not compositional and have no notion of observable behavior. In fact, they have not been defined with the aim of developing behavioural equivalences. In particular, [1] aims at simplifying the development of an interpreter of P Systems proved to be correct, [5] aims at proving the decidability of the divergence problem for the considered variant of P Systems, [6] aims at describing the causal dependencies occurring between applications of rules of a P System, and in [8] a formal framework is proposed to describe a large number of variants of P Systems.

The flattening result we obtain by considering P Systems without dissolving rules is similar to the result given in [3], where a notion of *computational encoding* is introduced and used to show that n -PBR Systems (PBR Systems with $n > 0$ membranes) can be simulated by 0-PBR Systems (PBR Systems with no membranes). We refer the reader to [4] for an introduction to PBR Systems. The difference between the result given in [3] and ours is that the axioms we give to transform a P System into its flat form is proved to preserve our compositional semantics, hence it is sound with respect to any behavioural equivalence. The flat system we obtain can replace the original one in any bigger system without changing the global behaviour.

Another normal form of P Systems is introduced in [9], where it is shown that any P System of grade k (namely, in which the depth of the membrane nesting tree is k) consisting of a composition of n membranes can be reduced to an equivalent P System of grade 2 with the same number of membranes. In this case the P System in normal form is equivalent to the original one in the sense that it can generate the same language, where a word of the language is the concatenation of the objects sent outside the skin membrane during the execution of the system. This means that the original system and the one in normal form can be considered as equivalent even if one of the two performs additional steps in which no objects are sent out of the skin. The notion of equivalence we consider here, instead, is stronger. In fact, in order to ensure that a system in normal form can always replace its original system in any context, we need to require that the two systems are step-by-step equivalent.

2 The P Algebra: Syntax and Semantics

In this section we recall the *P Algebra*, the algebraic notation of P Systems we have introduced in [2]. Constants of the P Algebra correspond to single objects or single evolution rules, and they can be composed into membrane systems by using operations of union, containment in a membrane, juxtaposition of membranes, and so on. Terms of the P Algebra are the states of the LTS.

We assume that objects belong to an alphabet V , and we assume the usual string notation to represent multisets of objects. For instance, to represent $\{a, a, b, b, c\}$ we may write either $aabbc$, or a^2b^2c , or $(ab)^2c$. We denote multiset (and set) union as string concatenation, hence we write u_1u_2 for $u_1 \cup u_2$. For the sake of readability, we shall write $u \rightarrow v_h v_o \{v_i\}$ for the generic non-dissolving evolution rule $u \rightarrow (v_h, here)(v_o, out)(v_1, in_{l_1}) \dots (v_n, in_{l_n})$, and $u \rightarrow v_h v_o \{v_i\} \delta$ for the similar generic dissolving evolution rule.

The abstract syntax of the P Algebra is defined as follows.

Definition 1 (P Algebra). *The abstract syntax of membrane contents c , membranes m , and membrane systems ms is given by the following grammar, where l ranges over \mathbb{N} and a over the set of object names V :*

$$\begin{aligned} c &::= (\emptyset, \emptyset) \mid (u \rightarrow v_h v_o \{v_i\}, \emptyset) \mid (u \rightarrow v_h v_o \{v_i\} \delta, \emptyset) \mid (\emptyset, a) \mid c \cup c \\ m &::= [l c]_l \\ ms &::= m \mid ms \mid ms \mid \mu(m, ms) \mid v \end{aligned}$$

A membrane content c represents a pair (\mathcal{R}, u) , where \mathcal{R} is a set of evolution rules and u is a multiset of objects. A membrane content is obtained through the union operation \cup from constants representing single evolution rules and constants representing single objects, and can be plugged into a membrane with label l by means of the operation $[l _]_l$ of membranes m . As a consequence, given a membrane content c representing the pair (\mathcal{R}, u) and $l \in \mathbb{N}$, $[l c]_l$ represents the membrane having l as label, \mathcal{R} as evolution rules and u as objects. For the sake of simplicity, we shall usually write $(\mathcal{R}_1 \mathcal{R}_2, u_1 u_2)$ for $(\mathcal{R}_1 u_1) \cup (\mathcal{R}_2, u_2)$, $[l u]_l$ for $[l (\emptyset, u)]_l$ and $[l u_1 \rightarrow v_{h1} v_{o1} \{v_{i1}\}, \dots, u_1 \rightarrow v_{hn} v_{on} \{v_{in}\}, u]_l$ for $[l (\mathcal{R}, u)]_l$ if $\mathcal{R} = \{u_1 \rightarrow v_{h1} v_{o1} \{v_{i1}\}, \dots, u_1 \rightarrow v_{hn} v_{on} \{v_{in}\}\}$.

Membrane systems ms have the following meaning: $ms_1 \mid ms_2$ represents the juxtaposition of ms_1 and ms_2 , $\mu(m, ms)$ represents the hierarchical composition of m and ms , namely the containment of ms in m , and v represents the *dissolved membrane*. Juxtaposition is used to group sibling membranes, namely membranes all having the same parent in a membrane structure. This operation allows hierarchical composition μ to be defined as a binary operator on a single membrane (the parent) and a juxtaposition of membrane (all the children) rather than on $n + 1$ membranes, for any possible number of children n . Finally, the dissolved membrane v will be used in the definition of the LTS to denote the state of a membrane after the application of one of its dissolving rules.

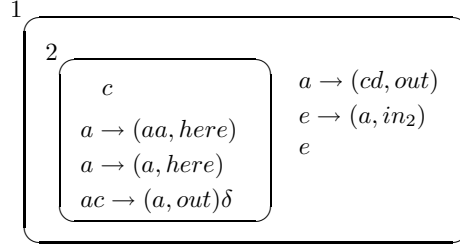


Fig. 1. An example of P System that may send out of the skin membrane a multiset of objects $c^n d^n$ for any $n \in \mathbb{N}$.

As an example, the P System shown in Figure 1 corresponds to the following membrane system:

$$\mu([{}_1 a \rightarrow (cd, out), e \rightarrow (a, in_2), e]_1, [{}_2 a \rightarrow (aa, here), a \rightarrow (a, here), ac \rightarrow (a, out)\delta, c]_2).$$

Moreover, a P System similar to the one shown in Figure 1, but in which membrane 1 contains also a membrane with label 3 containing, in turn, an object a and no rules, corresponds to the following membrane system:

$$\mu([{}_1 a \rightarrow (cd, out), e \rightarrow (a, in_2), e]_1, [{}_2 a \rightarrow (aa, here), a \rightarrow (a, here), ac \rightarrow (a, out)\delta, c]_2 \mid [{}_3 a]_3).$$

The semantics of the P Algebra is given in terms of an LTS, namely a triple $(\mathcal{S}, \mathcal{L}, \{\xrightarrow{\ell} \mid \ell \in \mathcal{L}\})$, where \mathcal{S} is a set of *states*, \mathcal{L} is a set of *labels*, and $\xrightarrow{\ell} \subseteq \mathcal{S} \times \mathcal{S}$ is a *transition relation* for each $\ell \in \mathcal{L}$. As usual, we write $s \xrightarrow{\ell} s'$ for $(s, s') \in \xrightarrow{\ell}$. LTS labels can be of the following forms:

- $(u, U, v, v', M, I, O^\uparrow, O^\downarrow)$, describing a computation step performed by a membrane content c , where:
 - u is the multiset of objects consumed by the application of evolution rules in c , as it results from the composition, by means of $_ \cup _$, of the constants representing these evolution rules.
 - U is the set of multisets of objects corresponding to the left hand sides of the evolution rules in c .
 - v is the multiset of objects in c offered for the application of the evolution rules, as it results from the composition, by means of $_ \cup _$, of the constants representing these objects. When operation $[_]_l$ is applied to c , it is required that v and u coincide.
 - v' is the multiset of objects in c that are not used to apply any evolution rule and, therefore, are not consumed, as it results from the composition, by means of $_ \cup _$, of the constants representing these objects. When operation

$[l_]l$ is applied to c , it is required that no multiset in U is contained in v' , thus implying that no evolution rule in c can be further applied by exploiting the available objects. This constraint is mandatory to ensure maximal parallelism.

- M contains a membrane label l if some evolution rule in c is not applied since its firing would imply sending objects to some child membrane labeled l , but no child membrane labeled l exists. When the operation μ is applied to $([l_]c, ms)$, for any membrane system ms and membrane label l' , it is required that l is not a membrane in ms .
- I is the multiset of objects received as input from the parent membrane and from the child membranes.
- O^\uparrow is the multiset of objects sent as an output to the parent membrane.
- O^\downarrow is a set of pairs (l_i, v_{l_i}) describing the multiset of objects sent as an output to each child membrane l_i .
- $(M, \mathcal{I}, O^\uparrow, O^\downarrow)$, describing a computation step performed by a membrane system ms , where: \mathcal{I} is a set of pairs (l_i, v_{l_i}) describing the multiset of objects received as an input by each membrane l_i in ms , and M, O^\uparrow and O^\downarrow are as in the previous case.

Components $I, O^\downarrow, O^\uparrow$ in labels of the first form, and components $\mathcal{I}, O^\downarrow, O^\uparrow$ in labels of the second form, describe the input/output behavior of P Algebra terms, namely what is usually considered to be the observable behavior. Labels of the first form are more complex since u, U, v, v' are needed to infer the behavior of membrane contents compositionally. For the same reason M is used in both forms of labels.

Now, LTS transitions are defined through SOS transition rules of the form $\frac{\text{premises}}{\text{conclusion}}$, where the premises are a set of transitions, and the conclusion is a transition. Intuitively, SOS transition rules permit us to infer moves of P Algebra terms from moves of their subterms. Rules of the semantics are given in Appendix A.

3 Flattening Systems without Dissolving Rules

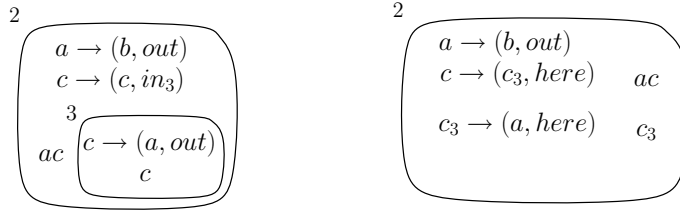


Fig. 2. An example of flattening of a P System without dissolving rules.

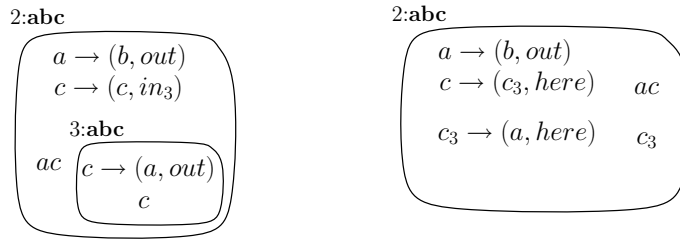


Fig. 3. An example of flattening of a P System without dissolving rules in which membranes are enriched with interfaces.

As shown in [3], any transition P System with a fixed membrane structure (i.e. without dissolving rules) can be reduced to a flat form in which the membrane structure consists only of one membrane. If we assume that membrane labels of a P System are unique, this result can be obtained by moving objects and rules of inner membranes into the external membrane, after suitable renaming. An example of application of this technique is shown in Figure 2. However, the behaviour of the flat membrane is the same as the behaviour of the original membrane structure only under the assumption that the membrane cannot receive any object from the external environment. In fact, if the external environment could send to the flat membrane an object that is the renaming of some object originally in an inner membrane, this could enable the application of some rules among those that have been added to the external membrane by the flattening technique. In the example of the figure, if the environment could send an object c_3 inside the membrane on the right, this would enable the application of rule $c_3 \rightarrow (a, here)$ which would result, after one more step, in the output of a b that would not be sent out by the original system.

To solve this problem we consider a slightly extended variant of P Systems in which each membrane is enriched with an interface, namely a set of objects representing the only objects that can be received from the environment. This means that if in the environment of a membrane there is a rule willing to send into it some objects that are not in the corresponding interface, then such a rule will never be applicable. Note that this extension is rather conservative, namely it is always possible to find a set of objects large enough to ensure that the behaviour of a P System extended with interfaces is the same as the intended behaviour of the original P System. As an example, in Figure 3 we extend the P Systems of Figure 2 with interfaces (placed together with membrane labels), and we obtain that in this case the behaviour of the two systems is really the same, as now the environment cannot send c_3 into the external membrane.

Now we formally define the syntax of the P Algebra extended with interfaces on membranes. The main difference with respect to the original syntax, given in Definition 1, is that the operation $[_]_l$ of membranes m is extended with a set of objects i . Moreover, since we aim at introducing a notion of flat membrane,

namely a membrane which cannot have inner membranes, we extend the syntax of the P Algebra also with a new operation $\llbracket_l _ \rrbracket_l^i$ denoting a membrane that cannot be used as the first operand of a $\mu(-, -)$ operation.

Definition 2 (P Algebra with Interfaces). *The abstract syntax of membrane contents c , membranes m , and membrane systems ms is given by the following grammar, where l ranges over \mathbb{N} , a over V and $i \subseteq V$:*

$$\begin{aligned} c &::= (\emptyset, \emptyset) \mid (u \rightarrow v_h v_o \{v_i\}, \emptyset) \mid (u \rightarrow v_h v_o \{v_i\} \delta, \emptyset) \mid (\emptyset, a) \mid c \cup c \\ m &::= [l c]_l^i \\ ms &::= m \mid ms \mid ms \mid \mu(m, ms) \mid \llbracket_l c \rrbracket_l^i \mid \mathbf{v} \end{aligned}$$

We also give the formal definition of the semantics of the P Algebra extended with interfaces on membranes. The main difference with respect to the original semantics is that the objects that can be received as an input by a membrane must belong to the interface of the membrane itself. Moreover, the new semantics has also to describe the behaviour of the new operation $\llbracket_l _ \rrbracket_l^i$. Formally, the SOS rules of the semantics of the P Algebra with Interfaces can be defined by starting from those given in Appendix A. In order to describe the behaviour of interfaces we replace rules (m1) and (m2) with the following four rules:

$$\frac{x \xrightarrow[u, U, u, v']{M, \emptyset, O^\uparrow, O^\downarrow} y \quad \delta \notin O^\uparrow}{[l x]_l^i \xrightarrow{M, \emptyset, O^\uparrow, O^\downarrow} [l y]_l^i} \quad (m1')$$

$$\frac{x \xrightarrow[u, U, u, v']{M, I, O^\uparrow, O^\downarrow} y \quad \delta \notin O^\uparrow \quad \text{Set}(I) \subseteq i \quad I \neq \emptyset}{[l x]_l^i \xrightarrow{M, (l, I), O^\uparrow, O^\downarrow} [l y]_l^i} \quad (m1'')$$

$$\frac{x \xrightarrow[u, U, u, v']{M, \emptyset, O^\uparrow, O^\downarrow} y \quad \delta \in O^\uparrow}{[l x]_l^i \xrightarrow{M, \emptyset, O^\uparrow, O^\downarrow} \mathbf{v}} \quad (m2')$$

$$\frac{x \xrightarrow[u, U, u, v']{M, I, O^\uparrow, O^\downarrow} y \quad \delta \in O^\uparrow \quad \text{Set}(I) \subseteq i \quad I \neq \emptyset}{[l x]_l^i \xrightarrow{M, (l, I), O^\uparrow, O^\downarrow} \mathbf{v}} \quad (m2'')$$

where $\text{Set}(I)$ is the *underlying set* of multiset I , namely the set of all the objects occurring in I . The rules (m1') and (m1'') replace the old rule (m1). Here we distinguish the case $I = \emptyset$ and the case $I \neq \emptyset$, since we prefer to have \emptyset instead of (l, \emptyset) in the label component showing inputs received from the environment. Finally, in order to describe the behaviour of flat membranes we add the following four rules:

$$\frac{x \xrightarrow[u, U, u, v']{M, \emptyset, O^\uparrow, \emptyset} y \quad \delta \notin O^\uparrow}{\llbracket l x \rrbracket_l^i \xrightarrow{M, \emptyset, O^\uparrow, \emptyset} \llbracket l y \rrbracket_l^i} \quad (fm1')$$

$$\frac{x \xrightarrow[u, U, u, v']{M, I, O^\uparrow, \emptyset} y \quad \delta \notin O^\uparrow \quad \text{Set}(I) \subseteq i \quad I \neq \emptyset}{\llbracket l x \rrbracket_l^i \xrightarrow{M, (I, I), O^\uparrow, \emptyset} \llbracket l y \rrbracket_l^i} \quad (fm1'')$$

$$\frac{x \xrightarrow[u, U, u, v']{M, \emptyset, O^\uparrow, \emptyset} y \quad \delta \in O^\uparrow}{\llbracket l x \rrbracket_l^i \xrightarrow{M, \emptyset, O^\uparrow, \emptyset} \mathbf{v}} \quad (fm2')$$

$$\frac{x \xrightarrow[u, U, u, v']{M, I, O^\uparrow, \emptyset} y \quad \delta \in O^\uparrow \quad \text{Set}(I) \subseteq i \quad I \neq \emptyset}{\llbracket l x \rrbracket_l^i \xrightarrow{M, (I, I), O^\uparrow, \emptyset} \mathbf{v}} \quad (fm2'')$$

Notice that rules for $\llbracket l x \rrbracket_l^i$ require that the multiset of objects sent to inner membranes (fourth component of the label) is empty.

The new semantics rules, similarly to all the other rules of the semantics, respect the constraints of the well-known *de Simone* format [7] which ensures that all the behavioural equivalences considered in [2] are congruences.

Now, the flattening technique for systems without dissolving rules can be expressed by means of axioms. We first give some basic axioms on the commutativity and associativity of the operations of the P Algebra and on simple properties of membranes with empty interfaces and of flat membranes.

Definition 3 (Basic axioms). *The basic axioms are the following:*

$$c_1 \cup c_2 = c_2 \cup c_1 \quad (\cup_1)$$

$$c_1 \cup (c_2 \cup c_3) = (c_1 \cup c_2) \cup c_3 \quad (\cup_2)$$

$$ms_1 \mid ms_2 = ms_2 \mid ms_1 \quad (|_1)$$

$$ms_1 \mid (ms_2 \mid ms_3) = (ms_1 \mid ms_2) \mid ms_3 \quad (|_2)$$

$$[l_1 c]_{l_1}^\emptyset = [l_2 c]_{l_2}^\emptyset \quad (if1)$$

$$ms = ms \mid [l \mathcal{R}, \emptyset]_l^\emptyset \quad (if2)$$

$$\llbracket l_1 c \rrbracket_{l_1}^\emptyset = \llbracket l_2 c \rrbracket_{l_2}^\emptyset \quad (if3)$$

$$ms = ms \mid \llbracket l \mathcal{R}, \emptyset \rrbracket_l^\emptyset \quad (if4)$$

$$\llbracket l_1 c \rrbracket_{l_1}^i = \mu \left([l_1 c]_{l_1}^i, [l_2 \mathcal{R}, \emptyset]_{l_2}^\emptyset \right) \quad (fm1)$$

$$\llbracket l_1 c \rrbracket_{l_1}^i = \mu \left([l_1 c]_{l_1}^i, \llbracket l_2 \mathcal{R}, \emptyset \rrbracket_{l_2}^\emptyset \right) \quad (fm2)$$

The first four axioms state commutativity and associativity of union of membrane contents and juxtaposition of membrane systems. Axiom *(if1)* states that if a membrane has empty interface then its name l_1 can be changed into l_2 . The reason is that l_1 cannot receive any object from any outer membrane, and any evolution rule sending objects to l_1 is never applicable. Axioms *(if2)* and *(fm1)* state that a membrane with no object and with empty interface can be juxtaposed with any membrane system or inserted inside another membrane, since its rules are never applicable. Axioms *(if3)*, *(if4)* and *(fm2)* are the same as *(if1)*, *(if2)* and *(fm1)*, respectively, but dealing with flat membranes.

The flattening technique we are going to define is based on renaming of objects. In the example of Figure 3, the object c contained in membrane 3 is renamed into c_3 when it is moved to membrane 2 in order to distinguish it from the other object c that occurs in membrane 2. Consequently, rules of membranes 2 and 3 have to be modified before merging them in membrane 2 resulting from flattening. For this reason, we define two functions **FlatIn** and **FlatOut**. The former gives the result of the renaming of the rules of the membrane that is removed by the flattening. The latter gives the result of the renaming of the rules of the membrane which contains the one that is removed. In order to avoid ambiguities, in the definitions of **FlatIn** and **FlatOut** we shall use the notation $u \rightarrow (v_h, here)(v_o, out)(v_{l_1}, in_{l_1}) \dots (v_{l_n}, in_{l_n})$ for evolution rules rather than the more compact notation $u \rightarrow v_h v_o \{v_{l_i}\}$.

We assume that the alphabet V is partitioned as follows: $V = \bar{V} \cup (\bigcup_{L \in \mathbb{N}^+} V_L)$, where \bar{V} is the set of all objects without subscripts and V_L is the set obtained by adding $L \in \mathbb{N}^+$ as a subscript to each object of \bar{V} . In other words, if $\bar{V} = a, b, c, \dots$, then $V_1 = a_1, b_1, c_1, \dots$, $V_{1.2} = a_{1.2}, b_{1.2}, c_{1.2}, \dots$ and so on. Moreover, let $\text{Rid}(u, l)$ denote the multiset obtained by replacing each occurrence of each object a_L in u with an occurrence of object $a_{L.l}$. For example, $\text{Rid}(abc_3, 3) = a_3 a_3 b_3 c_3 c_3 c_3 = a_3^2 b_3 c_3^3$, and $\text{Rid}(aa_1 bb_2, 3) = a_3 a_{1.3} b_3 b_{2.3}$. The functions **FlatIn** and **FlatOut** are defined as follows:

$$\begin{aligned} \text{FlatIn}(u \rightarrow (v_h, here)(v_o, out)(v_{l_1}, in_{l_1}) \dots (v_{l_n}, in_{l_n}), l) = \\ \text{Rid}(u, l) \rightarrow (\text{Rid}(v_h, l)v_o, here)(\emptyset, out)(v_{l_1}, in_{l_1}) \dots (v_{l_n}, in_{l_n}) \\ \text{FlatOut}(u \rightarrow (v_h, here)(v_o, out)(v_{l_1}, in_{l_1}) \dots (v_{l_i}, in_{l_i}) \dots (v_{l_n}, in_{l_n}), l_i) = \\ u \rightarrow (v_h \text{Rid}(v_{l_i}, l_i), here)(v_o, out)(v_{l_1}, in_{l_1}) \dots (\emptyset, in_{l_i}) \dots (v_{l_n}, in_{l_n}) \end{aligned}$$

Both **FlatIn** and **FlatOut** take a rule and a membrane label as arguments, and give a new rule as result. In both cases the membrane label represents the label of the membrane that is removed by the flattening. In the first case such a label (denoted l) should not occur in the evolution rule, as the rule is assumed to be one of those of the inner membrane involved in the flattening. In the second case the label certainly occurs in the evolution rule (in fact it is denoted l_i) as the rule is assumed to be one of those of the outer membrane involved in the flattening. With abuse of notation we shall write $\text{FlatIn}(\mathcal{R}, l)$ for $\{\text{FlatIn}(r, l) \mid r \in \mathcal{R}\}$, and $\text{FlatOut}(\mathcal{R}, l)$ for $\{\text{FlatOut}(r, l) \mid r \in \mathcal{R}\}$.

Now, the flattening technique is expressed by means of the following axioms.

Definition 4 (Flattening axioms). Let \mathcal{R}_1 and \mathcal{R}_2 be sets of evolution rules containing no dissolving rule, and let \mathcal{R}_1 and u_1 contain no objects in V_{L,l_2} . The flattening axioms are the following:

$$\frac{ms \neq v \quad V_{L,l_2} \cap i_1 = \emptyset \quad \text{FlatOut}(\mathcal{R}_1, l_2) = \mathcal{R}'_1 \quad \text{FlatIn}(\mathcal{R}_2, l_2) = \mathcal{R}'_2}{\mu([l_1 \mathcal{R}_1, u_1]_{l_1}^{i_1}, [l_2 \mathcal{R}_2, u_2]_{l_2}^{i_2} | ms) = \mu([l_1 \mathcal{R}'_1 \mathcal{R}'_2, u_1 \text{Rid}(u_2, l_2)]_{l_1}^{i_1}, ms)} \quad (f1)$$

$$\frac{ms \neq v \quad V_{L,l_2} \cap i_1 = \emptyset \quad \text{FlatOut}(\mathcal{R}_1, l_2) = \mathcal{R}'_1 \quad \text{FlatIn}(\mathcal{R}_2, l_2) = \mathcal{R}'_2}{\mu([l_1 \mathcal{R}_1, u_1]_{l_1}^{i_1}, [l_2 \mathcal{R}_2, u_2]_{l_2}^{i_2} | ms) = \mu([l_1 \mathcal{R}'_1 \mathcal{R}'_2, u_1 \text{Rid}(u_2, l_2)]_{l_1}^{i_1}, ms)} \quad (f2)$$

$$\frac{V_{L,l_2} \cap i_1 = \emptyset \quad \text{FlatOut}(\mathcal{R}_1, l_2) = \mathcal{R}'_1 \quad \text{FlatIn}(\mathcal{R}_2, l_2) = \mathcal{R}'_2}{\mu([l_1 \mathcal{R}_1, u_1]_{l_1}^{i_1}, [l_2 \mathcal{R}_2, u_2]_{l_2}^{i_2}) = \mu([l_1 \mathcal{R}'_1 \mathcal{R}'_2, u_1 \text{Rid}(u_2, l_2)]_{l_1}^{i_1})} \quad (f3)$$

$$\frac{V_{L,l_2} \cap i_1 = \emptyset \quad \text{FlatOut}(\mathcal{R}_1, l_2) = \mathcal{R}'_1 \quad \text{FlatIn}(\mathcal{R}_2, l_2) = \mathcal{R}'_2}{\mu([l_1 \mathcal{R}_1, u_1]_{l_1}^{i_1}, [l_2 \mathcal{R}_2, u_2]_{l_2}^{i_2}) = \mu([l_1 \mathcal{R}'_1 \mathcal{R}'_2, u_1 \text{Rid}(u_2, l_2)]_{l_1}^{i_1})} \quad (f4)$$

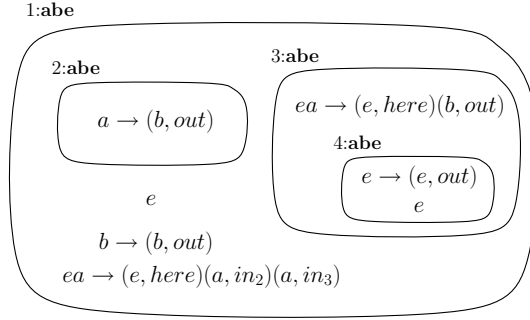


Fig. 4. An example of P System.

As an example, let us consider the P System in Figure 4 which corresponds to the P Algebra term $t = \mu(m_1, m_2 | \mu(m_3, m_4))$ where:

$$m_1 = [{}_1 \mathcal{R}_1, e]_1^{abe} \text{ with } \mathcal{R}_1 = \{ea \rightarrow (e, here)(a, in_2)(a, in_3), b \rightarrow (b, out)\}$$

$$m_2 = [{}_2 \mathcal{R}_2, \emptyset]_2^{abe} \text{ with } \mathcal{R}_2 = \{a \rightarrow (b, out)\}$$

$$m_3 = [{}_3 \mathcal{R}_3, \emptyset]_3^{abe} \text{ with } \mathcal{R}_3 = \{ea \rightarrow (e, here)(b, out)\}$$

$$m_4 = [{}_4 \mathcal{R}_4, e]_4^{abe} \text{ with } \mathcal{R}_4 = \{e \rightarrow (e, out)\} .$$

Now, we have that

$$\begin{aligned} \mu(m_3, m_4) &\stackrel{(f3)}{=} \llbracket_3 \text{FlatOut}(\mathcal{R}_3, 4) \text{FlatIn}(\mathcal{R}_4, 4), \emptyset \text{Rid}(e, 4) \rrbracket_3^{abe} \\ &= \llbracket_3 ea \rightarrow (e, \text{here})(b, \text{out}), e_4 \rightarrow (e, \text{here}), e_4 \rrbracket_3^{abe}. \end{aligned}$$

Let us denote with fm_3 the flat membrane we have obtained. Now, we can go on applying axioms as follows:

$$\begin{aligned} t &= \mu(m_1, m_2 \mid \mu(m_3, m_4)) = \mu(m_1, m_2 \mid fm_3) \\ &\stackrel{(f1)}{=} \mu(\llbracket_1 \text{FlatOut}(\mathcal{R}_1, 2) \text{FlatIn}(\mathcal{R}_2, 2), e \text{Rid}(\emptyset, 2) \rrbracket_1^{abe}, \\ &\quad \llbracket_3 ea \rightarrow (e, \text{here})(b, \text{out}), e_4 \rightarrow (e, \text{here}), e_4 \rrbracket_3^{abe}) \\ &= \mu(\llbracket_1 ea \rightarrow (ea_2, \text{here})(a, \text{in}_3), b \rightarrow (b, \text{out}), a_2 \rightarrow (b, \text{here}), e \rrbracket_1^{abe}, \\ &\quad \llbracket_3 ea \rightarrow (e, \text{here})(b, \text{out}), e_4 \rightarrow (e, \text{here}), e_4 \rrbracket_3^{abe}) \\ &\stackrel{(f4)}{=} \llbracket_1 ea \rightarrow (ea_2 a_3, \text{here}), b \rightarrow (b, \text{out}), a_2 \rightarrow (b, \text{here}), \\ &\quad e_3 a_3 \rightarrow (e_3, \text{here})(b, \text{out}), e_{43} \rightarrow (e_3, \text{here}), ee_{43} \rrbracket_1^{abe}. \end{aligned}$$

Proposition 1 (soundness). *The portions of the LTS that are rooted in terms equated by axioms (f1)–(f4) are isomorphic.*

Proof. We start with the proof for (f1). We prove that the portion of the LTS rooted in $\mu(\llbracket_{l_1} \mathcal{R}_1, u_1 \rrbracket_{l_1}^{i_1}, \llbracket_{l_2} \mathcal{R}_2, u_2 \rrbracket_{l_2}^{i_2} \mid ms)$ is isomorphic to a part of the portion of the LTS rooted in $\mu(\llbracket_{l_1} \mathcal{R}'_1 \mathcal{R}'_2, u_1 \text{Rid}(u_2, l_2) \rrbracket_{l_1}^{i_1}, ms)$. More precisely, we prove that, given any transition $\mu(\llbracket_{l_1} \mathcal{R}_1, u_1 \rrbracket_{l_1}^{i_1}, \llbracket_{l_2} \mathcal{R}_2, u_2 \rrbracket_{l_2}^{i_2} \mid ms) \xrightarrow{l} t$, for any term t , then there is a transition $\mu(\llbracket_{l_1} \mathcal{R}'_1 \mathcal{R}'_2, u_1 \text{Rid}(u_2, l_2) \rrbracket_{l_1}^{i_1}, ms) \xrightarrow{l} t'$ such that t and t' are equated by the same axiom (f1).

Take any transition from $\mu(\llbracket_{l_1} \mathcal{R}_1, u_1 \rrbracket_{l_1}^{i_1}, \llbracket_{l_2} \mathcal{R}_2, u_2 \rrbracket_{l_2}^{i_2} \mid ms)$. Such a transition must be inferred from a transition of each of its three components. These three transitions have the following shape:

$$\llbracket_{l_1} \mathcal{R}_1, u_1 \rrbracket_{l_1}^{i_1} \xrightarrow{M_1, \{(l_1, I_1)\}, O_1^\dagger, O_1^\dagger} \llbracket_{l_1} \mathcal{R}_1, u'_1 \rrbracket_{l_1}^{i_1} \quad (1)$$

$$\llbracket_{l_2} \mathcal{R}_2, u_2 \rrbracket_{l_2}^{i_2} \xrightarrow{M_2, \{(l_2, I_2)\}, O_2^\dagger, \emptyset} \llbracket_{l_2} \mathcal{R}_2, u'_2 \rrbracket_{l_2}^{i_2} \quad (2)$$

$$ms \xrightarrow{M, \mathcal{I}, O^\dagger, \emptyset} ms' \quad (3)$$

Then, transitions (2) and (3) originate transition

$$\llbracket_{l_2} \mathcal{R}_2, u_2 \rrbracket_{l_2}^{i_2} \mid ms \xrightarrow{\emptyset, \{(l_2, I_2)\} \mathcal{I}, O_2^\dagger O_1^\dagger, \emptyset} \llbracket_{l_2} \mathcal{R}_2, u'_2 \rrbracket_{l_2}^{i_2} \mid ms' \quad (4)$$

by semantic rule (*jux1*). The transition from $\mu(\llbracket_{l_1} \mathcal{R}_1, u_1 \rrbracket_{l_1}^{i_1}, \llbracket_{l_2} \mathcal{R}_2, u_2 \rrbracket_{l_2}^{i_2} \mid ms)$ is inferred through semantic rule (*h1*) from (1) and (4) and takes the shape:

$$\mu \left([l_1 \mathcal{R}_1, u_1]_{l_1}^{i_1}, [l_2 \mathcal{R}_2, u_2]_{l_2}^{i_2} \mid ms \right) \xrightarrow{\emptyset, \{(l_1, I_1 \setminus (O^\uparrow O_2^\dagger))\}, O_1^\dagger, \emptyset} \mu \left([l_1 \mathcal{R}_1, u'_1]_{l_1}^{i_1}, [l_2 \mathcal{R}_2, u'_2]_{l_2}^{i_2} \mid ms' \right) \quad (5)$$

provided that:

1. $O_1^\downarrow \simeq \mathcal{I} \cup \{(l_2, I_2)\}$;
2. $O^\uparrow O_2^\dagger \subseteq I_1$.

Note that the first constraint above implies that there exists some $O_1^{\downarrow'}$ such that

$$O_1^\downarrow = \{(l_2, I_2)\} \cup O_1^{\downarrow'} \quad (6)$$

Now, (1) can be inferred through $(m1')$ or $(m1'')$. The two cases are similar, let us assume the case $(m1')$. Analogously, let us assume that (2) is inferred through semantic rule $(m2')$. The two originating transitions have the shape:

$$(\mathcal{R}_1, u_1) \xrightarrow[v_1, U_1, v_1, v'_1]{M_1, I_1, O_1^\dagger, \{(l_2, I_2)\} \cup O_1^{\downarrow'}} (\mathcal{R}_1, u'_1) \quad (7)$$

$$(\mathcal{R}_2, u_2) \xrightarrow[v_2, U_2, v_2, v'_2]{M_2, I_2, O_2^\dagger, \emptyset} (\mathcal{R}_2, u'_2) \quad (8)$$

for suitable values $v_1, U_1, v'_1, v_2, U_2, v'_2$.

Notice that this implies that $\text{Set}(I_1) \subseteq i_1$. From (7) we infer

$$(\mathcal{R}'_1, u_1) \xrightarrow[v_1, U_1, v_1, v'_1]{M_1, I_1^\dagger, O_1^\dagger, O_1^{\downarrow'}} (\mathcal{R}'_1, u'_1 \setminus (\text{Rid}(I_2, l_2))) \quad (9)$$

By removing input O_2^\dagger we infer

$$(\mathcal{R}'_1, u_1) \xrightarrow[v_1, U_1, v_1, v'_1]{M_1, I_1 \setminus O_2^\dagger, O_1^\dagger, O_1^{\downarrow'}} (\mathcal{R}'_1, u'_1 \setminus (O_2^\dagger \text{Rid}(I_2, l_2))) \quad (10)$$

From (8) we infer:

$$(\mathcal{R}'_2, \text{Rid}(u_2, l_2)) \xrightarrow[v_2, \hat{U}_2, v_2, v'_2]{M_2, \emptyset, \emptyset, \emptyset} (\mathcal{R}'_2, \text{Rid}(u'_2 \setminus I_2, l_2) O_2^\dagger) \quad (11)$$

where $\hat{v}_2, \hat{U}_2, \hat{v}'_2$ denote $\text{Rid}(v_2, l_2)$, $\text{Rid}(U_2, l_2)$, $\text{Rid}(v'_2, l_2)$, respectively.

Through semantic rule $(u1)$, from (10) and (11) we infer

$$(\mathcal{R}'_1 \mathcal{R}'_2, u_1 \text{Rid}(u_2, l_2)) \xrightarrow[v_1 \hat{v}_2, U_1 \oplus \hat{U}_2, v_1 \hat{v}_2, v'_1 \hat{v}'_2]{M_1 M_2, I_1 \setminus O_2^\dagger, O_1^\dagger, O_1^{\downarrow'}} (\mathcal{R}'_1 \mathcal{R}'_2, u'_1 \text{Rid}(u'_2, l_2)) \quad (12)$$

By applying semantic rule $(m1)$, which is applicable since $\text{Set}(I_1) \subseteq i_1$, we infer:

$$[l_1 \mathcal{R}'_1 \mathcal{R}'_2, u_1 \text{Rid}(u_2, l_2)]_{l_1}^{i_1} \xrightarrow{M_1 M_2, \{(l_1, I_1 \setminus O_2^\uparrow)\}, O_1^\uparrow, O_1^\downarrow} [l_1 \mathcal{R}'_1 \mathcal{R}'_2, u'_1 \text{Rid}(u'_2, l_2)]_{l_1}^{i_1} \quad (13)$$

We already know that $O_1^\downarrow \simeq \mathcal{I} \cup \{(l_2, I_2)\}$, $O^\uparrow O_2^\uparrow \subseteq I_1$ and $O_1^\downarrow = \{(l_2, I_2)\} \cup O_1^\downarrow$. Therefore, $O_1^\downarrow \simeq \mathcal{I}$ and $O^\uparrow \subseteq I_1 \setminus O_2^\uparrow$. So, we can apply the semantic rule (h1) to infer that (3) and (13) originate

$$\begin{aligned} & \mu([l_1 \mathcal{R}'_1 \mathcal{R}'_2, u_1 \text{Rid}(u_2, l_2)]_{l_1}^{i_1}, ms) \\ & \xrightarrow{\emptyset, \{(l_1, I_1 \setminus O^\uparrow O_2^\uparrow)\}, O_1^\uparrow, \emptyset} \\ & \mu([l_1 \mathcal{R}'_1 \mathcal{R}'_2, u'_1 \text{Rid}(u'_2, l_2)]_{l_1}^{i_1}, ms') . \quad (14) \end{aligned}$$

Summarizing, we have proved that if we take any transition from term $\mu([l_1 \mathcal{R}_1, u_1]_{l_1}^{i_1}, [l_2 \mathcal{R}_2, u_2]_{l_2}^{i_2} \mid ms)$ we have a corresponding transition from term $\mu([l_1 \mathcal{R}'_1 \mathcal{R}'_2, u_1 \text{Rid}(u_2, l_2)]_{l_1}^{i_1}, ms)$, where the two transitions have the same label and have terms related by axiom (f1) in the right side.

The converse is similar, with the use of premise $V_{L \cdot l_2} \cap i_1 = \emptyset$.

The proof of the case of axiom (f2) is the same as the one of axiom (f1), but for minor differences in transition labels. Moreover, the cases of axioms (f3) and (f4) are analogous to the case of (f1) and (f2), respectively, thanks to basic axioms that allow us to rewrite the term $\llbracket l_1 \mathcal{R}'_1 \mathcal{R}'_2, u_1 \text{Rid}(u_2, l_2) \rrbracket_{l_1}^{i_1}$ as the term $\mu([l_1 \mathcal{R}'_1 \mathcal{R}'_2, u_1 \text{Rid}(u_2, l_2)]_{l_1}^{i_1}, [l \emptyset]_l^\emptyset)$ and $\mu([l_1 \mathcal{R}_1, u_1]_{l_1}^{i_1}, \llbracket l_2 \mathcal{R}_2, u_2 \rrbracket_{l_2}^{i_2})$ as the term $\mu([l_1 \mathcal{R}_1, u_1]_{l_1}^{i_1}, \llbracket l_2 \mathcal{R}_2, u_2 \rrbracket_{l_2}^{i_2} \mid [l \emptyset]_l^\emptyset)$.

Theorem 1. *Any membrane system $\mu(m, ms)$ with multisets of objects and evolution rules built over \bar{V} , without dissolving rules and with membranes having unique labels, can be reduced to an equivalent flat membrane.*

Proof. This can be proved by induction on the number of membrane nodes in the membrane nesting tree of $\mu(m, ms)$. If the membrane nesting tree contains two membranes, namely $\mu(m, ms)$ is either $\mu([l_1 c_1]_{l_1}, [l_2 c_2]_{l_2})$ or $\mu([l_1 c_1]_{l_1}, \llbracket l_2 c_2 \rrbracket_{l_2})$, then the proof follows immediately from axioms (f3) and (f4). If the membrane nesting tree contains more than two membranes, then the proof can be done by resorting to the induction hypothesis after applying one of the axioms (f1), (f2), (f3) and (f4) to one of the leaves of the tree.

Since the size of a term is always finite (and consequently the membrane nesting tree is finite) the flat form is reached after a finite number of steps. The facts that no dissolving rules are present, that multisets of objects and evolution rules are built by using objects from \bar{V} and that membranes are labeled with unique labels, ensure that the assumptions and the premises of the axioms are always satisfied. Finally, Proposition 1 ensures that all the applications of the flattening axioms preserve the behaviour, hence the behaviour of the final flat membrane is equivalent to the one of the original membrane system.

4 Problems that Arise with Dissolving Rules and Possible Solutions

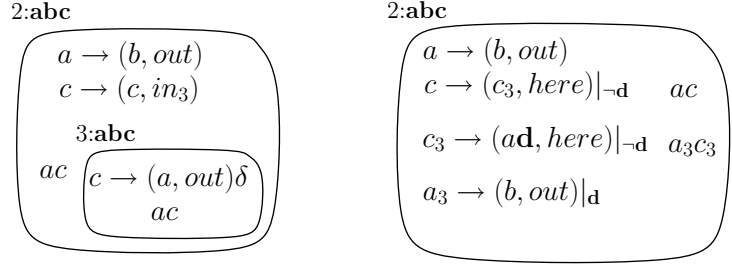


Fig. 5. Example of flattening in which the inner membrane contains a dissolving rule.

In the membrane obtained by our flattening technique, dissolution of a membrane contained in another one has to be simulated. In general, when a dissolving rule is applied in a membrane, we have that (i) the objects of such a membrane become immediately available to the outer membrane, (ii) rules of such a membrane disappear, and (iii) rules of the outer membrane which send objects to the membrane that has been dissolved become no longer applicable.

One possible way of simulating dissolution is by replacing δ with a special object \mathbf{d} in every dissolving rule and using such a special object as a promoter or inhibitor of some rules obtained by the flattening (after extending the syntax and the semantics of the P Algebra to deal with promoters and inhibitors). This would allow (ii) and (iii) to be simulated by using \mathbf{d} as an inhibitor of those rules obtained by the flattening and corresponding to the rules of the dissolved membrane and to the rules sending objects to the dissolved membranes. Moreover, (i) can be simulated by defining the flattening in such a way that the rules of the outer membrane are copied with the objects they consume renamed, in order to allow such rules to be applied to the objects representing the objects of the dissolved membrane. These new rules should have \mathbf{d} as a promoter.

We give a simple example of flattening with dissolution of inner membranes in Figure 5. Here, the rule causing dissolution of membrane 3 is rewritten into a new rule having objects renamed as described in the previous section and producing \mathbf{d} . Now, both the rule originally in 2 and sending objects to 3, and the rule originally in 3 require that \mathbf{d} has not yet been produced. Moreover, a new rule promoted by \mathbf{d} has been introduced to simulate that the objects originally in membrane 3 are available in membrane 2 after its dissolution.

The flattening technique explained in the previous section cannot be applied if the outer membrane contains a dissolving rule. As an example, let us consider Fig. 6, where flattening is applied. We can provide a context in which the original membrane system and the flat membrane behave differently (see Fig. 7). The point

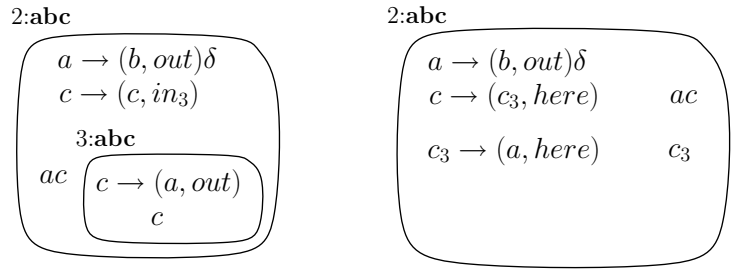


Fig. 6. Example of flattening in which the outer membrane contains a dissolving rule.

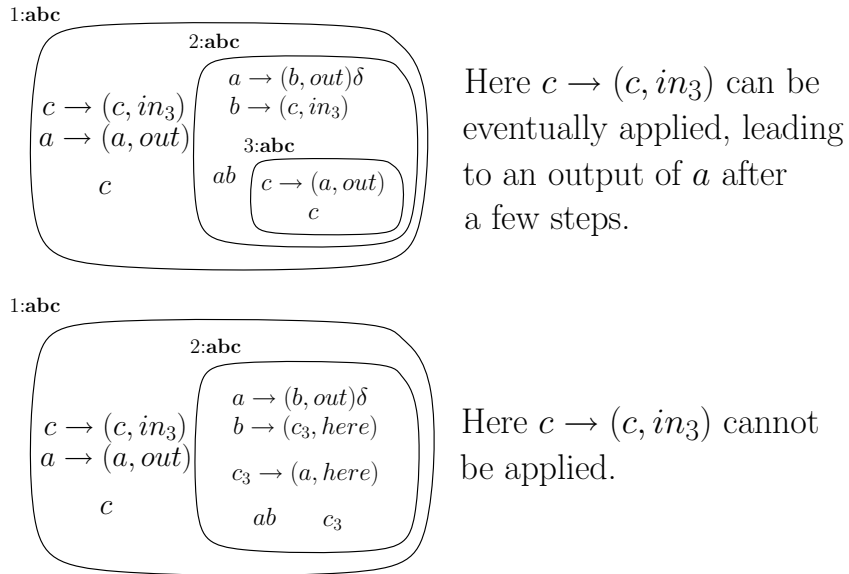


Fig. 7. Example of context in which the two membranes systems in Figure 6 behave differently.

is that the rule of membrane 1 sending object c to membrane 3 can be eventually applied if and only if membrane 3 still exists after the dissolution of membrane 2. In this case the only possible solution is to avoid flattening of membrane structures in which the outermost membrane can be dissolved. As a consequence, a general normal form for P Systems will have two shapes:

- If the external membrane of a membrane structure cannot be dissolved its normal form is a single flat membrane that cannot be dissolved.
- If the external membrane of a membrane structure can be dissolved its normal form is a structure consisting only of membranes that can be dissolved, but for innermost membranes that might be non-dissolvable.

5 Conclusions and Future Work

We have faced the problem of defining a flattening technique for P Systems defined by means of axioms on terms of the algebra of such systems we have introduced in [2], the P Algebra, and preserving the semantics. We have formally defined such a technique in the case of P Systems without dissolving rules. This has required extending the syntax and the semantics of the P Algebra with a notion of interface and with a notion of flat membrane, defining some axioms and proving that these axioms preserve the semantics. We have discussed the problems that arise when dissolving rules are taken into account, and we have proposed some possible solutions to these problems.

Our long term aim is to define a normal form of P System. In order to reach the normal form of a P System, in addition to apply our flattening technique we would also need to transform rules and objects of such a system into some minimal form. We believe that, given two systems in normal form, it will be possible to check their equivalence as follows:

- if they are both flat, they should contain the same rules and objects, up to a suitable renaming;
- if they are both non flat because the external membrane contain a dissolving rule (see Section 4), they should have the same membrane structure of equivalent membranes.

Acknowledgments

This research has been partially supported by MiUR PRIN 2006 Project “Biologically Inspired Systems and Calculi and their Applications (BISCA)”. We thank Pierluigi Frisco for interesting discussions.

References

1. O. Andrei, G. Ciobanu, D. Lucanu. A Rewriting Logic Framework for Operational Semantics of Membrane Systems. *Theoret. Comp. Sci.* 373 (2007) 163–181.
2. R. Barbuti, A. Maggiolo-Schettini, P. Milazzo, S.Tini. Compositional Semantics and Behavioral Equivalences for P Systems. *Theoret. Comput. Sci.*, in press.
3. L. Bianco, V. Manca. Encoding–Decoding Transitional Systems for Classes of P Systems. *Workshop on Membrane Computing (WMC 2005)*, LNCS 3850, pp. 134–143, Springer, 2006.
4. L. Bianco, F. Fontana, G. Franco, V. Manca. P Systems for Biological Dynamics. In: *Applications of Membrane Computing*, Springer, Berlin, 2006.
5. N. Busi. Using Well-structured Transition Systems to Decide Divergence for Catalytic P Systems. *Theoret. Comput. Sci.* 372 (2007) 125–135.
6. N. Busi. Causality in Membrane Systems. *Workshop on Membrane Computing (WMC 2007)*, LNCS 4860, pp. 160–171, Springer, 2007.

7. R. de Simone. High Level Synchronization Devices in Meije-SCCS. Theoret. Comput. Sci. 37, pp. 245–267, 1985.
8. R. Freund, S. Verlan. A Formal Framework for Static (Tissue) P Systems. Workshop on Membrane Computing (WMC 2007), LNCS 4860, pp. 271–284, Springer, 2007.
9. I. Petre. A Normal Form for P-Systems. Bulletin of the EATCS 67 (1999) 165–172.

A Rules of the Operational Semantics

In this section we recall the rules of the operational semantics of the P Algebra given in [2].

A.1 Rules for membrane contents

$$\frac{I \in V^* \quad n \in \mathbb{N}}{(u \rightarrow v_h v_o \{v_i\}, \emptyset) \xrightarrow[\frac{\emptyset, I, v_o^n, \{(l_i, v_i^n)\}}{u^n, \{u\}, \emptyset, \emptyset}]{\emptyset, I, v_o^n, \{(l_i, v_i^n)\}}} (u \rightarrow v_h v_o \{v_i\}, I v_h^n) \quad (mc1_n)$$

$$\frac{I \in V^* \quad n \in \mathbb{N} \quad n > 0}{(u \rightarrow v_h v_o \{v_i\} \delta, \emptyset) \xrightarrow[\frac{\emptyset, I, v_o^n v_h^n \delta, \{(l_i, v_i^n)\}}{u^n, \{u\}, \emptyset, \emptyset}]{\emptyset, I, v_o^n v_h^n \delta, \{(l_i, v_i^n)\}}} \mathbf{v} \quad (mc2_n)$$

$$\frac{I \in V^*}{(u \rightarrow v_h v_o \{v_i\} \delta, \emptyset) \xrightarrow[\frac{\emptyset, \{u\}, \emptyset, \emptyset}{\emptyset, I, \emptyset, \emptyset}]{\emptyset, I, \emptyset, \emptyset}} (u \rightarrow v_h v_o \{v_i\} \delta, I) \quad (mc3)$$

$$\frac{I \in V^* \quad M \subseteq \text{Labels}(\{v_i\}) \quad M \neq \emptyset}{(u \rightarrow v_h v_o \{v_i\}, \emptyset) \xrightarrow[\frac{\emptyset, \emptyset, \emptyset, \emptyset}{M, I, \emptyset, \emptyset}]{M, I, \emptyset, \emptyset}} (u \rightarrow v_h v_o \{v_i\}, I) \quad (mc4)$$

$$\frac{I \in V^* \quad M \subseteq \text{Labels}(\{v_i\}) \quad M \neq \emptyset}{(u \rightarrow v_h v_o \{v_i\} \delta, \emptyset) \xrightarrow[\frac{\emptyset, \emptyset, \emptyset, \emptyset}{M, I, \emptyset, \emptyset}]{M, I, \emptyset, \emptyset}} (u \rightarrow v_h v_o \{v_i\} \delta, I) \quad (mc5)$$

$$\frac{I \in V^*}{(\emptyset, a) \xrightarrow[\frac{\emptyset, \emptyset, a, \emptyset}{\emptyset, I, \emptyset, \emptyset}]{\emptyset, I, \emptyset, \emptyset}} (\emptyset, I) \quad (mc6) \qquad \frac{I \in V^*}{(\emptyset, a) \xrightarrow[\frac{\emptyset, \emptyset, \emptyset, a}{\emptyset, I, \emptyset, \emptyset}]{\emptyset, I, \emptyset, \emptyset}} (\emptyset, I a) \quad (mc7)$$

$$\frac{I \in V^*}{(\emptyset, \emptyset) \xrightarrow[\frac{\emptyset, \emptyset, \emptyset, \emptyset}{\emptyset, I, \emptyset, \emptyset}]{\emptyset, I, \emptyset, \emptyset}} (\emptyset, I) \quad (mc8)$$

A.2 Rules for union of membrane contents

$$\frac{x_1 \frac{M_1, I_1, O_1^\dagger, O_1^\downarrow}{u_1, U_1, v_1, v_1'} \rightarrow y_1 \quad x_2 \frac{M_2, I_2, O_2^\dagger, O_2^\downarrow}{u_2, U_2, v_2, v_2'} \rightarrow y_2 \quad \begin{array}{l} M_1 M_2 \cap \text{Labels}(O_1^\downarrow \cup_{\mathbb{N}} O_2^\downarrow) = \emptyset \\ v_1' v_2' \not\prec U_1 \oplus U_2 \quad \delta \notin O_1^\dagger O_2^\dagger \end{array}}{x_1 \cup x_2 \xrightarrow[u_1 u_2, U_1 \oplus U_2, v_1 v_2, v_1' v_2']{M_1 M_2, I_1 I_2, O_1^\dagger O_2^\dagger, O_1^\downarrow \cup_{\mathbb{N}} O_2^\downarrow} y_1 \cup y_2} \quad (u1)$$

$$\frac{x_1 \frac{M_1, I_1, O_1^\dagger, O_1^\downarrow}{u_1, U_1, v_1, v_1'} \rightarrow y_1 \quad x_2 \frac{M_2, I_2, O_2^\dagger, O_2^\downarrow}{u_2, U_2, v_2, v_2'} \rightarrow y_2 \quad \begin{array}{l} M_1 M_2 \cap \text{Labels}(O_1^\downarrow \cup_{\mathbb{N}} O_2^\downarrow) = \emptyset \\ v_1' v_2' \not\prec U_1 \oplus U_2 \quad \delta \in O_1^\dagger \quad \delta \notin O_2^\dagger \end{array}}{x_1 \cup x_2 \xrightarrow[u_1 u_2, U_1 \oplus U_2, v_1 v_2, v_1' v_2']{M_1 M_2, I_1 I_2, O_1^\dagger O_2^\dagger \text{Objects}(y_2), O_1^\downarrow \cup_{\mathbb{N}} O_2^\downarrow} \mathbf{v}} \quad (u2)$$

$$\frac{x_1 \frac{M_1, I_1, O_1^\dagger, O_1^\downarrow}{u_1, U_1, v_1, v_1'} \rightarrow y_1 \quad x_2 \frac{M_2, I_2, O_2^\dagger, O_2^\downarrow}{u_2, U_2, v_2, v_2'} \rightarrow y_2 \quad \begin{array}{l} M_1 M_2 \cap \text{Labels}(O_1^\downarrow \cup_{\mathbb{N}} O_2^\downarrow) = \emptyset \\ v_1' v_2' \not\prec U_1 \oplus U_2 \quad \delta \in O_1^\dagger \cap O_2^\dagger \end{array}}{x_1 \cup x_2 \xrightarrow[u_1 u_2, U_1 \oplus U_2, v_1 v_2, v_1' v_2']{M_1 M_2, I_1 I_2, O_1^\dagger O_2^\dagger, O_1^\downarrow \cup_{\mathbb{N}} O_2^\downarrow} \mathbf{v}} \quad (u3)$$

A.3 Rules for single membranes and juxtaposition of membranes

$$\frac{x \frac{M, I, O^\dagger, O^\downarrow}{u, U, u, v'} \rightarrow y \quad \delta \notin O^\dagger}{[i x]_l \xrightarrow{M, \{(l, I)\}, O^\dagger, O^\downarrow} [i y]_l} \quad (m1) \quad \frac{x \frac{M, I, O^\dagger, O^\downarrow}{u, U, u, v'} \rightarrow y \quad \delta \in O^\dagger}{[i x]_l \xrightarrow{M, \{(l, I)\}, O^\dagger, O^\downarrow} \mathbf{v}} \quad (m2)$$

$$\frac{x_1 \xrightarrow{M_1, \mathcal{I}_1, O_1^\dagger, \emptyset} y_1 \quad x_2 \xrightarrow{M_2, \mathcal{I}_2, O_2^\dagger, \emptyset} y_2 \quad \delta \notin O_1^\dagger O_2^\dagger}{x_1 | x_2 \xrightarrow{\emptyset, \mathcal{I}_1 \mathcal{I}_2, O_1^\dagger O_2^\dagger, \emptyset} y_1 | y_2} \quad (jux1)$$

$$\frac{x_1 \xrightarrow{M_1, \mathcal{I}_1, O_1^\dagger, \emptyset} y_1 \quad x_2 \xrightarrow{M_2, \mathcal{I}_2, O_2^\dagger, \emptyset} y_2 \quad \delta \in O_1^\dagger, \delta \notin O_2^\dagger}{x_1 | x_2 \xrightarrow{\emptyset, \mathcal{I}_1 \mathcal{I}_2, (O_1^\dagger O_2^\dagger) - \delta, \emptyset} y_2} \quad (jux2)$$

$$\frac{x_1 \xrightarrow{M_1, \mathcal{I}_1, O_1^\dagger, \emptyset} y_1 \quad x_2 \xrightarrow{M_2, \mathcal{I}_2, O_2^\dagger, \emptyset} y_2 \quad \delta \in O_1^\dagger \cap O_2^\dagger}{x_1 | x_2 \xrightarrow{\emptyset, \mathcal{I}_1 \mathcal{I}_2, (O_1^\dagger O_2^\dagger), \emptyset} \mathbf{v}} \quad (jux3)$$

A.4 Rules for hierarchy of membranes

$$\frac{x_1 \xrightarrow{M_1, \{(l_1, I_1)\}, O_1^\downarrow, O_1^\downarrow} y_1 \quad x_2 \xrightarrow{M_2, \mathcal{I}_2, O_2^\uparrow, \emptyset} y_2 \quad \begin{array}{l} O_1^\downarrow \simeq \mathcal{I}_2 \quad O_2^\uparrow \subseteq I_1 \\ M_1 \cap \mathbf{Labels}(\mathcal{I}_2) = \emptyset \quad \delta \notin O_1^\uparrow O_2^\uparrow \end{array}}{\mu(x_1, x_2) \xrightarrow{\emptyset, \{(l_1, I_1 \setminus O_2^\uparrow)\}, O_1^\uparrow, \emptyset} \mu(y_1, y_2)} \quad (h1)$$

$$\frac{x_1 \xrightarrow{M_1, \{(l_1, I_1)\}, O_1^\uparrow, O_1^\downarrow} y_1 \quad x_2 \xrightarrow{M_2, \mathcal{I}_2, O_2^\uparrow, \emptyset} y_2 \quad \begin{array}{l} O_1^\downarrow \simeq \mathcal{I}_2 \quad O_2^\uparrow \subseteq I_1 \quad \delta \in O_1^\uparrow \\ M_1 \cap \mathbf{Labels}(\mathcal{I}_2) = \emptyset \quad \delta \notin O_2^\uparrow \end{array}}{\mu(x_1, x_2) \xrightarrow{\emptyset, \{(l_1, I_1 \setminus O_2^\uparrow)\}, O_1^\uparrow - \delta, \emptyset} y_2} \quad (h2)$$

$$\frac{x_1 \xrightarrow{M_1, \{(l_1, I_1)\}, O_1^\uparrow, O_1^\downarrow} y_1 \quad x_2 \xrightarrow{M_2, \mathcal{I}_2, O_2^\uparrow, \emptyset} y_2 \quad \begin{array}{l} O_1^\downarrow \simeq \mathcal{I}_2 \quad O_2^\uparrow - \delta \subseteq I_1 \quad \delta \notin O_1^\uparrow \\ M_1 \cap \mathbf{Labels}(\mathcal{I}_2) = \emptyset \quad \delta \in O_2^\uparrow \end{array}}{\mu(x_1, x_2) \xrightarrow{\emptyset, \{(l_1, I_1 \setminus O_2^\uparrow)\}, O_1^\uparrow, \emptyset} y_1} \quad (h3)$$

$$\frac{x_1 \xrightarrow{M_1, \{(l_1, I_1)\}, O_1^\uparrow, O_1^\downarrow} y_1 \quad x_2 \xrightarrow{M_2, \mathcal{I}_2, O_2^\uparrow, \emptyset} y_2 \quad \begin{array}{l} O_1^\downarrow \simeq \mathcal{I}_2 \quad O_2^\uparrow - \delta \subseteq I_1 \\ M_1 \cap \mathbf{Labels}(\mathcal{I}_2) = \emptyset \quad \delta \in O_1^\uparrow \cap O_2^\uparrow \end{array}}{\mu(x_1, x_2) \xrightarrow{\emptyset, \{(l_1, I_1 \setminus O_2^\uparrow)\}, O_1^\uparrow, \emptyset} \mathbf{v}} \quad (h4)$$