Fundamenta Informaticae XX (2008) 1–34
IOS Press

A P Systems Flat Form Preserving Step–by–step Behaviour

Roberto Barbuti, Andrea Maggiolo-Schettini, Paolo Milazzo
Dipartimento di Informatica, Università di Pisa
Largo Pontecorvo 3, 56127 Pisa, Italy
Email: {barbuti, maggiolo, milazzo}@di.unipi.it

Simone Tini
Dip. di Scienze della Cultura, Politiche e dell'Informazione
Università dell'Insubria
Via Carloni 78, 22100 Como, Italy
Email: simone.tini@uninsubria.it

Abstract. 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 normal form with an equivalent semantics. As a first step, we introduce axioms which allow the transformation of membrane structures into flat membranes. 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 [13] as a labeled transition system (LTS) [3]. The class of P Systems we have considered are the so called *transition P Systems* with dissolving rules and with cooperative evolution rules. In the definition of the semantics, P Systems are seen as *reactive systems* [10], 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

Address for correspondence: Insert here address for correspondence

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 could eventually send some objects back to the environment, namely to outer and inner membranes.

The LTS we have defined allows us to observe the behaviour 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 membrane, and the multisets 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 [3] 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 transforms 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 equivalences we have given in [3]. It is not easy to prove soundness and completeness of an axiomatization, namely that axioms relate behaviourally 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 some axioms, that any P System in this class can be transformed into an equivalent P System consisting of only one 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 an external membrane.

We also consider P Systems with dissolving rules, and we show, by giving one more axiom, that any

P System in this class whose outmost membrane does not contain any dissolving rule, can be transformed into a flat system. In order to obtain this result we need to consider P Systems in which evolution rules may have promoters and inhibitors.

These results and the flattening technique we define in order to obtain them, can be used not only in the context of the definition of an axiomatization of our semantics. For instance, one interested in developing some analysis or verification technique on P Systems could develop such a technique on flat systems and use our flattening method to compile general systems into flat ones.

The results we obtain in this paper about the equivalence of a P System and of its corresponding flat form, are based on the notion of *bisimulation* [11, 12]. A bisimulation is an equivalence relation on states of an LTS that relates states from which transitions with the same labels can be performed, reaching states that are again related by the bisimulation. This means that bisimilar states of an LTS are able to simulate each other step–by–step.

We defined bisimulation for our semantics of P Systems in [3]. In this case, states of the LTS correspond to states of the computation of a P System, and transitions correspond to computation steps. The fact that we use bisimulation to prove the equivalence of a P System with its corresponding flat form means that we prove that the flat form preserves the behaviour of the P System at each computation step. We remark that bisimulation on P Systems is a stronger behavioural equivalence than the commonly used ones based on the comparison of the generated/accepted languages.

This paper is structured as follows. In Section 2 we recall the definition of P Systems and of the syntax and the semantics of the P Algebra (the algebraic notation of P Systems we introduced in [3]). In Section 3 we give some basic axioms on terms of the P Algebra. In Sections 4 and 5 we give our flattening technique for P Systems without and with dissolving rules, respectively. Finally, in Section 6 we draw some conclusions and discuss possible future developments.

Related work Operational semantics for P Systems have been proposed in [2, 6, 7, 9]. All these semantics are not compositional and have no notion of observable behaviour. In fact, they have not been defined with the aim of developing behavioural equivalences. In particular, [2] aims at simplifying the development of an interpreter of P Systems proved to be correct, [6] aims at proving the decidability of the divergence problem for the considered variant of P Systems, [7] aims at describing the causal dependencies occurring between applications of rules of a P System, and in [9] 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 [4], 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 [5] for an introduction to PBR Systems. The difference between the result given in [4] and ours is that the axioms we give to transform a P System into its flat form are proved to preserve our compositional semantics, hence they are 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 [14], where it is shown that any P System of degree 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 degree 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. P Systems and the P Algebra

In this section we recall the definition of P Systems [13] and of the P Algebra [3]. The class of P Systems we consider includes cooperative evolution rules, dissolving rules and rule promoters and inhibitors. Moreover, we extend this class of P Systems with a notion of membrane interface. We shall adapt the syntax and semantics of the P Algebra with respect to [3] to deal with these features.

2.1. P Systems

A P System consists of a *hierarchy of membranes* that do not intersect, with a distinguishable membrane, called the *skin membrane*, surrounding them all. As usual, we assume membranes to be labeled by natural numbers. Given a set of objects V, a membrane m contains a multiset of *objects* in V, a set of *evolution rules*, and possibly other membranes, called *child* membranes (m is also called the *parent* of its child membranes). Objects represent molecules swimming in a chemical solution, and evolution rules represent chemical reactions that may occur inside the membrane containing them. For each evolution rule there is a multiset of objects representing the reactants, and a multiset of objects representing the products of the chemical reaction. A rule in a membrane m can be applied only to objects in m, meaning that the reactants should be precisely in m, and not in its child membranes. The rule must contain target indications, specifying the membranes where the new objects produced by applying the rule are sent. The new objects either remain in m, or can be sent out of m, or can be sent into one of its child membranes, precisely identified by its label. Formally, the products of a rule are denoted with a multiset of *messages* of the following forms:

- $(v, here)$, meaning that the multiset of objects v produced by the rule remain in the same membrane m;
- (v, out), meaning that the multiset of objects v produced by the rule are sent out of m;
- (v, in_l), meaning that the multiset of objects v produced by the rule are sent into the child membrane l.

An evolution rule may have some *promoters* and some *inhibitors*. Promoters are objects that are required to be present and inhibitors are objects that are required to be absent in the membrane m in order to enable the application of the rule. Promoters will be denoted simply as objects, namely $a, b, c, \ldots$, while inhibitors will be denoted as objects preceded by a negation symbol, namely $\neg a, \neg b, \neg c, \ldots$. Given a set of promoter and inhibitor symbols D, we denote with D^+ and D^- the sets of objects containing all the objects occurring in D as promoters and all the objects occurring in D as inhibitors, respectively. We remark that D^+ and D^- are sets of objects, hence elements on D^- will not be preceded by $\neg$. Moreover, with $\neg D$ we denote the set obtained by transforming each promoter in D into an inhibitor and viceversa. As an example, if $D = \{a, \neg b, \neg c, d\}$ we have $D^+ = \{a, d\}$, $D^- = \{b, c\}$ and $\neg D = \{\neg a, b, c, \neg d\}$.

We can assume that all evolution rules have the following form, where $\{l_1, \ldots, l_n\}$ is a set of membrane labels in $\mathbb{N}$ and D is a set of promoters and inhibitors:

$$u \to (v_h, here)(v_o, out)(v_1, in_{l_1}) \ldots (v_n, in_{l_n})|_D \,.$$

An evolution rule in a membrane m is called *dissolving* if its application causes the disappearance of m. In this case, the objects in m and the child membranes of m remain free in the parent membrane of m, and the evolution rules of m are lost. The skin membrane cannot be dissolved. A dissolving evolution rule is denoted by adding to the products the special object δ such that $\delta \notin V$:

$$u \to (v_h, here)(v_o, out)(v_1, in_{l_1}) \ldots (v_n, in_{l_n})\delta|_D \,.$$

Application of evolution rules is done with maximal parallelism, namely at each evolution step a multiset of instances of evolution rules is chosen non–deterministically such that no other rule can be applied to the system obtained by removing all the objects necessary to apply all the chosen rules. The application of rules consists of removing all the reactants of the chosen rules from the system, adding the products of the rules by taking into account the target indications, and dissolving all the membranes in which a δ message has been produced. Promoters are not consumed by the application of the corresponding evolution rule. A presence of a single occurrence of a promoter can enable the application of more than one rule in each maximally parallel evolution step. Similarly, the presence of a single occurrence of a inhibitor forbids the application of all the evolution rules in which it appears.

A P System has a tree–structure in which the skin membrane is the root and the membranes containing no other membranes are the leaves. The only change to the structure that may happen is the removal of some nodes of the tree (apart from the root) caused by some δ object produced by evolution rules. Hence, we assume membranes labels to be unique: they are assigned at the beginning of the evolution by counting the membranes encountered during a breadth-first visit of the tree–structure, with 1 as the label of the skin membrane.

Up to now we described features of P Systems, such as promoters and inhibitors, already known in the literature [13]. In order to define a general flat form for P Systems we need an additional new feature of P Systems, namely *membrane interfaces*. We assume that each membrane of a P System is enriched with an interface, namely a set of objects representing the only objects that can be received from an external membrane. 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. On the other hand, it is possible to statically check whether each rule of a P System might eventually become applicable or not because of interfaces. As a consequence, if we assume that the skin membrane of a P System cannot receive any object from the external environment, then a P System with interfaces can be simulated by a P System without interfaces in which the latter is obtained from the former by removing those rules that would never become applicable because of interfaces. However, in the following we will construct P Systems compositionally, and in this case we cannot assume that the outmost membrane does not receive any object from the external environment. In the following we will use interfaces for both the description and the flattening of P Systems.

Now, we formally define P Systems.

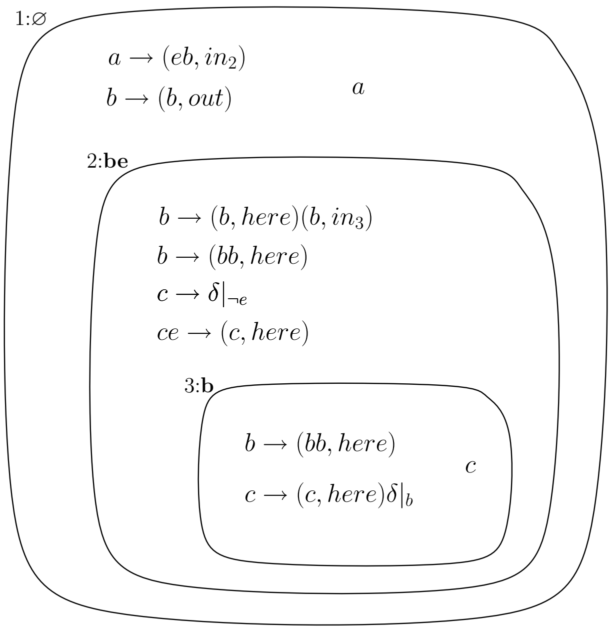

Figure 1. Example of P System that may send out of the skin membrane a multiset of objects b^{2^n} for any $n > 3$.

Definition 2.1. A *P System* Π is given by

$$\Pi = (V, \mu, i_1, \ldots, i_n, w_1, \ldots, w_n, R_1, \ldots, R_n)$$

where:

- V is an *alphabet* whose elements are called *objects*;
- $\mu \subset \mathbb{N} \times \mathbb{N}$ is a *membrane structure*, such that $(l_1, l_2) \in \mu$ denotes that the membrane labeled by l_2 is contained in the membrane labeled by l_1;
- i_j with $1 \leq j \leq n$ are subsets of V representing the *interfaces* of the membranes $1, \ldots, n$ of μ;
- w_j with $1 \leq j \leq n$ are strings from V^* representing multisets over V associated with the membranes $1, \ldots, n$ of μ;
- R_j with $1 \leq j \leq n$ are finite sets of *evolution rules* associated with the membranes $1, \ldots, n$ of μ.

We show in Figure 1 an example of P System in which all the main features of the formalism are used. In the figure, membranes are depicted as boxes containing evolution rules, objects and inner membranes. The label and the interface of a membrane are at a corner of the corresponding box separated by a colon.

2.2. The P Algebra: Syntax and Semantics

In this section we recall the *P Algebra*, the algebraic notation of P Systems we have introduced in [3]. Actually, we give here an extended version of the P Algebra which will allow us to transform arbitrary systems to an equivalent flat form. Such an extended version includes promoters and inhibitors in evolution rules and membrane interfaces.

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 the usual string notation to represent multisets of objects in V. For instance, to represent $\{a, a, b, b, c\}$ we may write either $aabbc$, or a^2b^2c, or $(ab)^2c$. We denote with $\mathsf{Set}(u)$ the support of multiset u, namely the set of all the objects occurring in u. 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 $u \rightarrow v_hv_o\{v_{l_i}\}|_D$ for the generic non–dissolving evolution rule $u \rightarrow (v_h, here)(v_o, out)(v_1, in_{l_1}) \ldots (v_n, in_{l_n})|_D$, and $u \rightarrow v_hv_o\{v_{l_i}\}\delta|_D$ for the similar generic dissolving evolution rule.

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

Definition 2.2. (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 $\mathbb{N}$, a over V and $i \subseteq V$:

$$\begin{aligned} c \ &::= (\varnothing, \varnothing) \ \big| \ (u \rightarrow v_hv_o\{v_{l_i}\}|_D, \varnothing) \ \big| \ (u \rightarrow v_hv_o\{v_{l_i}\}\delta|_D, \varnothing) \ \big| \ (\varnothing, a) \ \big| \ c \cup c \\ m \ &::= [_l\, c\,]_l^i \\ ms \ &::= ms \mid ms \ \big| \ \mu(m, ms) \ \big| \ F(m) \ \big| \ \mathsf{v} \end{aligned}$$

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 and interface i by means of the operation $[_l\, \_\,]_l^i$ of membranes m. As a consequence, given a membrane content c representing the pair $(\mathcal{R}, u)$, $l \in \mathbb{N}$ and $i \subseteq V$, $[_l\, c\,]_l^i$ represents the membrane having l as label, i as interface, $\mathcal{R}$ as evolution rules and u as objects.

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, $F(m)$ represents a *flat membrane*, namely it states that m does not contain any child membrane, 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 membranes (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.

In what follows we will often write $[\![_l\, c\,]\!]_l^i$ for $F([_l\, c\,]_l^i)$. For the sake of simplicity, we shall also often write $(\mathcal{R}_1\mathcal{R}_2, u_1u_2)$ for $(\mathcal{R}_1\, u_1) \cup (\mathcal{R}_2, u_2)$, $[_l\, u\,]_l^i$ for $[_l\, (\varnothing, u)\,]_l^i$ and $[_l\, u_1 \rightarrow v_{h1}v_{o1}\{v_{l_i1}\}|_{D_1}, \ldots, u_1 \rightarrow v_{hn}v_{on}\{v_{l_in}\}|_{D_n}, u\,]_l^i$ for $[_l\, (\mathcal{R}, u)\,]_l^i$ if $\mathcal{R} = \{u_1 \rightarrow v_{h1}v_{o1}\{v_{l_i1}\}|_{D_1}, \ldots, u_1 \rightarrow v_{hn}v_{on}\{v_{l_in}\}|_{D_n}\}$. We shall use analogous notations with $[_l\, \_\,]_l^i$ replaced by $[\![_l\, \_\,]\!]_l^i$.

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

$$\mu(\ [_1\, a \to (eb, in_2)\ ,\ b \to (b, out)\ ,\ a\,]_1^{\varnothing}\ ,\\ \mu(\ [_2\, b \to (b, here)(b, in_3)\ ,\ b \to (bb, here)\ ,\ c \to \delta|_{\neg e}\ ,\ ce \to (c, here)\ ]_2^{be}\ ,\\ [\![_3\, b \to (bb, here)\ ,\ c \to (c, here)\delta|_b\ ,\ c\,]\!]_3^{b})\ )\ .$$

We give another example in which juxtaposition of membranes is used. Consider a P System similar to the one shown in Figure 1, but in which membrane 2 contains also a membrane with label 4 containing, in turn, an object a and no rules. Such a P System corresponds to the following membrane system:

$$\mu(\ [_1\, a \to (eb, in_2)\ ,\ b \to (b, out)\ ,\ a\,]_1^{\varnothing}\ ,\\ \mu(\ [_2\, b \to (b, here)(b, in_3)\ ,\ b \to (bb, here)\ ,\ c \to \delta|_{\neg e}\ ,\ ce \to (c, here)\ ]_2^{be}\ ,\\ [\![_3\, b \to (bb, here)\ ,\ c \to (c, here)\delta|_b\ ,\ c\,]\!]_3^{b}\ \mid\ [\![_4\, a\,]\!]_4^{a}\ )\ )\ .$$

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:

- $(u, U, v, v', D, M, I, O^{\uparrow}, O^{\downarrow})$, 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 $[_l\ \_\,]_l^i$ 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 $[_l\ \_\,]_l^i$ 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.
  - D is a set of promoters and inhibitors required to be present and absent, respectively, by the application of evolution rules in c. More precisely, D^- contains all the inhibitors of the applied evolution rules in c, whereas D^+ is a subset of the promoters of those rules. Such a subset contains only those objects that are not present in the multiset of objects of c.
  - M is a set of pairs (l, a) where l is a membrane label and a is an object. M contains a pair (l, a) if some evolution rule in c is not applied since its firing would imply sending a to some child membrane labeled l, but either no child membrane labeled l exists, or a is not present in the interface of the child membrane labeled l.

– 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.

- $(M, \mathcal{I}^{\downarrow}, I^{\uparrow}, O^{\uparrow}, O^{\downarrow})$, describing a computation step performed by a membrane m, where: $\mathcal{I}^{\downarrow}$ is a set containing only the pair (l, I) where l is the label of m and I is the multiset of objects received by m as input from the parent membrane, $I^{\uparrow}$ is the multisets of objects received from the child membranes of m, and M, $O^{\uparrow}$ and $O^{\downarrow}$ are as in the previous case.
- $(\mathcal{I}^{\downarrow}, O^{\uparrow})$, describing a computation step performed by a membrane system ms, where $O^{\uparrow}$ is as in the previous cases and $\mathcal{I}^{\downarrow}$ differs with respect to the previous case because, when ms is a juxtaposition of membranes, it contains a pair (l, I) for each element of the juxtaposition.

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

For the sake of legibility, in transitions with labels of the first form we shall write the first five elements of the label below the arrow denoting the transition and the other four elements over the arrow.

Now, LTS transitions are defined through SOS transition rules [15] of the form $\frac{\textit{premises}}{\textit{conclusion}}$, where the premises are a set of transitions, and the conclusion is a transition. Intuitively, SOS transition rules permit us to infer transitions performed by P Algebra terms from transitions performed by their subterms. We assume the standard way to associate a set of transitions with a set of transition rules [1].

The transition rules we give in this paper differ slightly from those in [3] as we are describing the semantics of a version of the P Algebra extended with promoters, inhibitors and membrane interfaces. We give a brief explanation of the transition rules by focusing in particular on the differences with respect to [3]. We refer to that paper for a more detailed explanation of the semantics.

We start by giving the transition rules for membrane contents.

$$\frac{I \in V^* \qquad n \in \mathbb{N} \qquad D \cap \neg D = \varnothing}{(u \to v_h v_o \{v_{l_i}\}|_D, \varnothing) \xrightarrow[u^n, \{u\}, \varnothing, \varnothing, D]{\varnothing, I, v_o^n, \{(l_i, v_{l_i}^n)\}} (u \to v_h v_o \{v_{l_i}\}|_D, I v_h^n)} \qquad (mc1_n)$$

$$\frac{I \in V^* \qquad n \in \mathbb{N} \qquad n > 0 \qquad D \cap \neg D = \varnothing}{(u \to v_h v_o \{v_{l_i}\}\delta|_D, \varnothing) \xrightarrow[u^n, \{u\}, \varnothing, \varnothing, D]{\varnothing, I, I v_o^n v_h^n \delta, \{(l_i, v_{l_i}^n)\}} \mathsf{v}} \qquad (mc2_n)$$

$$\frac{I \in V^*}{(u \to v_h v_o \{v_{l_i}\}\delta|_D, \varnothing) \xrightarrow[\varnothing, \{u\}, \varnothing, \varnothing, \varnothing]{\varnothing, I, \varnothing, \varnothing} (u \to v_h v_o \{v_{l_i}\}\delta|_D, I)} \qquad (mc3)$$

$$\frac{I \in V^* \qquad l \in \mathsf{Labels}(\{v_{l_i}\}) \qquad a \in v_l}{(u \to v_h v_o \{v_{l_i}\}|_D, \varnothing) \xrightarrow[\varnothing, \varnothing, \varnothing, \varnothing, \varnothing]{\{(l,a)\}, I, \varnothing, \varnothing} (u \to v_h v_o \{v_{l_i}\}|_D, I)} \qquad (mc4)$$

$$\frac{I \in V^* \qquad l \in \mathsf{Labels}(\{v_{l_i}\}) \qquad a \in v_l}{(u \rightarrow v_h v_o\{v_{l_i}\}\delta|_D, \varnothing) \xrightarrow[\varnothing,\varnothing,\varnothing,\varnothing,\varnothing]{\{(l,a)\},I,\varnothing,\varnothing} (u \rightarrow v_h v_o\{v_{l_i}\}\delta|_D, I)} \quad (mc5)$$

$$\frac{I \in V^* \qquad D' \subseteq \neg D \qquad D' \neq \varnothing \qquad D' \cap \neg D' = \varnothing}{(u \rightarrow v_h v_o\{v_{l_i}\}|_D, \varnothing) \xrightarrow[\varnothing,\varnothing,\varnothing,\varnothing,D']{\varnothing,I,\varnothing,\varnothing} (u \rightarrow v_h v_o\{v_{l_i}\}|_D, I)} \quad (mc6)$$

$$\frac{I \in V^* \qquad D' \subseteq \neg D \qquad D' \neq \varnothing \qquad D' \cap \neg D' = \varnothing}{(u \rightarrow v_h v_o\{v_{l_i}\}\delta|_D, \varnothing) \xrightarrow[\varnothing,\varnothing,\varnothing,\varnothing,D']{\varnothing,I,\varnothing,\varnothing} (u \rightarrow v_h v_o\{v_{l_i}\}\delta|_D, I)} \quad (mc7)$$

$$\frac{I \in V^* \qquad D \cap \neg D \neq \varnothing}{(u \rightarrow v_h v_o\{v_{l_i}\}|_D, \varnothing) \xrightarrow[\varnothing,\varnothing,\varnothing,\varnothing,\varnothing]{\varnothing,I,\varnothing,\varnothing} (u \rightarrow v_h v_o\{v_{l_i}\}|_D, I)} \quad (mc8)$$

$$\frac{I \in V^* \qquad D \cap \neg D \neq \varnothing}{(u \rightarrow v_h v_o\{v_{l_i}\}\delta|_D, \varnothing) \xrightarrow[\varnothing,\varnothing,\varnothing,\varnothing,\varnothing]{\varnothing,I,\varnothing,\varnothing} (u \rightarrow v_h v_o\{v_{l_i}\}\delta|_D, I)} \quad (mc9)$$

$$\frac{I \in V^*}{(\varnothing, a) \xrightarrow[\varnothing,\varnothing,a,\varnothing,\varnothing]{\varnothing,I,\varnothing,\varnothing} (\varnothing, I)} \quad (mc10) \qquad \frac{I \in V^*}{(\varnothing, a) \xrightarrow[\varnothing,\varnothing,\varnothing,a,\varnothing]{\varnothing,I,\varnothing,\varnothing} (\varnothing, Ia)} \quad (mc11)$$

$$\frac{I \in V^*}{(\varnothing, \varnothing) \xrightarrow[\varnothing,\varnothing,\varnothing,\varnothing,\varnothing]{\varnothing,I,\varnothing,\varnothing} (\varnothing, I)} \quad (mc12)$$

Rule $(mc1_n)$ describes n simultaneous applications of a non–dissolving evolution rule for any $n \in \mathbb{N}$, and $(mc2_n)$ describes n simultaneous applications of a dissolving evolution rule with $n > 0$. In these two cases the set of promoters and inhibitors D of the applied evolution rule become part of the label of the transition. This permits the subsequent verification of the presence (absence) of the promoters (inhibitors) in the whole membrane content in which the rule is applied. The set D is checked to be coherent (the same object should not appear at the same time as promoter and as inhibitor) by the requirement $D \cap \neg D = \varnothing$. We could omit this checking, provided that the property is assured for all evolution rules. The use of rule $(mc1_n)$ with $n = 0$ describes the case in which promoters and inhibitors in D are assumed to be present and absent, respectively, but the rule is not applied because there are no objects u available in the whole membrane content in which the rule occurs. The non–application of a dissolving rule for the same reason is described by rule $(mc3)$.

Rules $(mc4)$ and $(mc5)$ describe the case in which a non–dissolving and a dissolving evolution rule, respectively, are not applied because they require sending objects to a child membrane l that is either assumed not to exist, or it is assumed to have an interface that does not include one of the sent objects, denoted a. In these rules we assume a function Labels from sets of pairs $\{(v_1, in_{l_1}), \ldots, (v_n, in_{l_n})\}$ with $v_i \neq \varnothing$, sets usually denoted $\{v_{l_i}\}$, into sets of labels. The function $\mathsf{Labels}(\{v_{l_i}\})$ extracts all the membrane labels from the given set of pairs, namely it returns $\{l_1, \ldots, l_n\}$.

Rules $(mc6)$ and $(mc7)$ describe the case in which a non–dissolving and a dissolving evolution rule, respectively, are not applied because some of the promoters and inhibitors in D they require to be present and absent, respectively, are assumed not to satisfy the requirements. In this case, such promoters and inhibitors are used as label of the defined transition after negation, namely $D' \subseteq \neg D$, because we will subsequently need to verify that the involved promoters and inhibitors are absent and present, respectively, that is the opposite of what verified when the evolution rule is applied. As in $(mc1_n)$ and $(mc2_n)$, the set of promoters and inhibitors used as transition label D' is checked to be coherent.

Rules $(mc8)$ and $(mc9)$ describe the case in which a non–dissolving and a dissolving evolution rule, respectively, are not applied because they have an incoherent set of promoters and inhibitors. In this case the membrane content can only receive objects from the environment.

Rules $(mc10), (mc11)$ and $(mc12)$ describe the transitions performed by membrane contents consisting of a single object and the transitions performed by an empty membrane content.

Now we give the transition rules allowing us to infer the behaviour of unions of membrane contents from the behaviour of the individual membrane contents.

$$\frac{\begin{array}{c} x_1 \xrightarrow[u_1,U_1,v_1,v_1',D_1]{M_1,I_1,O_1^{\uparrow},O_1^{\downarrow}} y_1 \qquad x_2 \xrightarrow[u_2,U_2,v_2,v_2',D_2]{M_2,I_2,O_2^{\uparrow},O_2^{\downarrow}} y_2 \qquad \delta \notin O_1^{\uparrow}O_2^{\uparrow} \\ v_1'v_2' \nvdash U_1 \oplus U_2 \qquad \nexists u \in V^*.((l,a) \in M_1M_2 \wedge (l,au) \in O_1^{\downarrow} \cup_{\mathbb{N}} O_2^{\downarrow}) \\ (D_1^- \cup D_2^-) \cap \mathsf{Set}(v_1v_1'v_2v_2') = \varnothing \qquad D_1 \cap \neg D_2 = \varnothing \qquad D = (D_1D_2) \setminus \mathsf{Set}(v_1v_1'v_2v_2') \end{array}}{x_1 \cup x_2 \xrightarrow[u_1u_2,U_1\oplus U_2,v_1v_2,v_1'v_2',D]{M_1M_2,I_1I_2,O_1^{\uparrow}O_2^{\uparrow},O_1^{\downarrow}\cup_{\mathbb{N}}O_2^{\downarrow}} y_1 \cup y_2} \quad (u1)$$

$$\frac{\begin{array}{c} x_1 \xrightarrow[u_1,U_1,v_1,v_1',D_1]{M_1,I_1,O_1^{\uparrow},O_1^{\downarrow}} y_1 \qquad x_2 \xrightarrow[u_2,U_2,v_2,v_2',D_2]{M_2,I_2,O_2^{\uparrow},O_2^{\downarrow}} y_2 \qquad \delta \in O_1^{\uparrow} \quad \delta \notin O_2^{\uparrow} \\ v_1'v_2' \nvdash U_1 \oplus U_2 \qquad \nexists u \in V^*.((l,a) \in M_1M_2 \wedge (l,au) \in O_1^{\downarrow} \cup_{\mathbb{N}} O_2^{\downarrow}) \\ (D_1^- \cup D_2^-) \cap \mathsf{Set}(v_1v_1'v_2v_2') = \varnothing \qquad D_1 \cap \neg D_2 = \varnothing \qquad D = (D_1D_2) \setminus \mathsf{Set}(v_1v_1'v_2v_2') \end{array}}{x_1 \cup x_2 \xrightarrow[u_1u_2,U_1\oplus U_2,v_1v_2,v_1'v_2',D]{M_1M_2,I_1I_2,O_1^{\uparrow}O_2^{\uparrow}\mathsf{Objects}(y_2),O_1^{\downarrow}\cup_{\mathbb{N}}O_2^{\downarrow}} \mathsf{v}} \quad (u2)$$

$$\frac{\begin{array}{c} x_1 \xrightarrow[u_1,U_1,v_1,v_1',D_1]{M_1,I_1,O_1^{\uparrow},O_1^{\downarrow}} y_1 \qquad x_2 \xrightarrow[u_2,U_2,v_2,v_2',D_2]{M_2,I_2,O_2^{\uparrow},O_2^{\downarrow}} y_2 \qquad \delta \in O_1^{\uparrow} \cap O_2^{\uparrow} \\ v_1'v_2' \nvdash U_1 \oplus U_2 \qquad \nexists u \in V^*.((l,a) \in M_1M_2 \wedge (l,au) \in O_1^{\downarrow} \cup_{\mathbb{N}} O_2^{\downarrow}) \\ (D_1^- \cup D_2^-) \cap \mathsf{Set}(v_1v_1'v_2v_2') = \varnothing \qquad D_1 \cap \neg D_2 = \varnothing \qquad D = (D_1D_2) \setminus \mathsf{Set}(v_1v_1'v_2v_2') \end{array}}{x_1 \cup x_2 \xrightarrow[u_1u_2,U_1\oplus U_2,v_1v_2,v_1'v_2',D]{M_1M_2,I_1I_2,O_1^{\uparrow}O_2^{\uparrow},O_1^{\downarrow}\cup_{\mathbb{N}}O_2^{\downarrow}} \mathsf{v}} \quad (u3)$$

Rules $(u1), (u2)$ and $(u3)$ describe the behaviour of a union of membrane contents when in none, in the first and in both of the composed membrane contents a dissolving rule is applied. We omit the symmetric rule of $(u2)$ where in the second membrane content a dissolving rule is applied. In these transition rules we use some auxiliary notations. Given a multiset of objects u and a set of multisets of objects U, we write $u \vdash U$ if there exists some $u' \subseteq u$ such that $u' \in U$, and we write $u \nvdash U$ otherwise. Intuitively, since for each $u' \in U$ there is an evolution rule having u' in the left side, then $u \vdash U$ means that u can let at least one of these transitions fire. Moreover, we assume a function $\mathsf{Objects}$ from membrane contents to multisets of objects such that $\mathsf{Objects}((\mathcal{R}, u)) = u$. Then, given two sets of multisets U_1 and U_2, we write $U_1 \oplus U_2$ to denote the set $\{u \in U_1U_2 | \nexists u' \in U_1U_2.u' \subset u\}$. Finally, given two sets $O_1^{\downarrow}$ and $O_2^{\downarrow}$ representing two outputs to inner membranes, we write $O_1^{\downarrow} \cup_{\mathbb{N}} O_2^{\downarrow}$ to denote the set $\{(l, uv) \,|\, (l, u) \in O_1^{\downarrow} \wedge (l, v) \in O_2^{\downarrow}\} \cup \{(l, u) \,|\, (l, u) \in O_1^{\downarrow} \wedge \nexists v.(l, v) \in O_2^{\downarrow}\} \cup \{(l, v) \,|\, (l, v) \in O_2^{\downarrow} \wedge \nexists u.(l, u) \in O_1^{\downarrow}\}$.

In $(u1), (u2)$ and $(u3)$ it is checked that the information represented by the pairs in M_1M_2 is coherent with the information on the objects sent to inner membranes represented by $O_1^{\downarrow} \cup_{\mathbb{N}} O_2^{\downarrow}$. As regards promoters and inhibitors, in $(u1), (u2)$ and $(u3)$ it is checked that none of the inhibitors in D_1 and D_2 occurs in the multisets of objects of the composed membrane contents, namely $(D_1^- \cup D_2^-) \cap \mathsf{Set}(v_1 v_1' v_2 v_2') = \varnothing$, and that there is no conflict between the requirements of the composed membrane contents, namely $D_1 \cap \neg D_2 = \varnothing$. Moreover, the label of the transition performed by the union of the membrane contents contains as set of promoters and inhibitors the set obtained by removing from the union of D_1 and D_2 the promoters corresponding to objects that occur in any of the two membrane contents.

Now we give the transition rules for individual membranes.

$$\frac{x \xrightarrow[u,U,u,v',D]{M,I,O^\uparrow,O^\downarrow} y \quad \delta \notin O^\uparrow \quad D^+ = \varnothing}{[_l\, x\,]_l^i \xrightarrow{M,\varnothing,I,O^\uparrow,O^\downarrow} [_l\, y\,]_l^i} \qquad (m1)$$

$$\frac{x \xrightarrow[u,U,u,v',D]{M,I_1 I_2,O^\uparrow,O^\downarrow} y \quad \delta \notin O^\uparrow \quad D^+ = \varnothing \quad \mathsf{Set}(I_1) \subseteq i \quad I_1 \neq \varnothing}{[_l\, x\,]_l^i \xrightarrow{M,\{(l,I_1)\},I_2,O^\uparrow,O^\downarrow} [_l\, y\,]_l^i} \qquad (m2)$$

$$\frac{x \xrightarrow[u,U,u,v',D]{M,I,O^\uparrow,O^\downarrow} y \quad \delta \in O^\uparrow \quad D^+ = \varnothing}{[_l\, x\,]_l^i \xrightarrow{M,\varnothing,I,O^\uparrow,O^\downarrow} \mathsf{v}} \qquad (m3)$$

$$\frac{x \xrightarrow[u,U,u,v',D]{M,I_1 I_2,O^\uparrow,O^\downarrow} y \quad \delta \in O^\uparrow \quad D^+ = \varnothing \quad \mathsf{Set}(I_1) \subseteq i \quad I_1 \neq \varnothing}{[_l\, x\,]_l^i \xrightarrow{M,\{(l,I_1)\},I_2,O^\uparrow,O^\downarrow} \mathsf{v}} \qquad (m4)$$

Rules $(m1)$ and $(m2)$ describe the transitions performed by a membrane with label l and interface i when no dissolving rule is applied. In particular, $(m1)$ describes the case in which no objects are received as an input from the external membrane, while $(m2)$ describes the case in which a multiset of objects $I_1 \neq \varnothing$ are received. In $(m2)$ the objects received as an input from the external membrane must be mentioned in the interface of membrane l, namely $\mathsf{Set}(I_1) \subseteq i$. Notice that, given I, all possible partitions of I into I_1 and I_2 must be considered. Rules $(m3)$ and $(m4)$ are analogous to $(m1)$ and $(m2)$ but describe the transitions performed when a dissolving rule is applied.

In all these transition rules the set of promoters and inhibitors D in the label of the transition performed by the membrane content must contain no promoters, namely $D^+ = \varnothing$. This ensures that all the promoters required by the evolution rules applied in the described computation step are present because they have all been removed by occurrences of $(u1), (u2)$ and $(u3)$ in the derivation tree of the transition performed by the membrane content. Moreover, it is also ensured that no object occurs in the membrane content which is an inhibitor of evolution rules applied in the described computation step. This follows from the definition of transition rules $(u1), (u2)$ and $(u3)$ used to derive the transition of the membrane content.

Now we give the transition rules that allow us to infer the behaviour of a flat membrane $[\![_l\, c\,]\!]_l^i = F([_l\, c\,]_l^i)$ from the behaviour of membrane $[_l\, c\,]_l^i$.

$$\frac{x \xrightarrow{M,\mathcal{I}^\downarrow,\varnothing,O^\uparrow,\varnothing} y \qquad \delta \notin O^\uparrow}{F(x) \xrightarrow{\mathcal{I}^\downarrow,O^\uparrow} F(y)} \qquad (fm1) \qquad\qquad \frac{x \xrightarrow{M,\mathcal{I}^\downarrow,\varnothing,O^\uparrow,\varnothing} y \qquad \delta \in O^\uparrow}{F(x) \xrightarrow{\mathcal{I}^\downarrow,O^\uparrow} \mathsf{v}} \qquad (fm2)$$

Notice that rules for $F(x)$ only require that both the multiset of objects received from and sent to inner membranes (third and fifth component of the label, respectively) are empty.

Now we give transition rules allowing us to infer the behaviour of juxtapositions of membranes from the behaviour of their components.

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

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

$$\frac{x_1 \xrightarrow{\mathcal{I}_1, O_1^{\uparrow}} y_1 \quad x_2 \xrightarrow{\mathcal{I}_2, O_2^{\uparrow}} y_2 \quad \delta \in O_1^{\uparrow} \cap O_2^{\uparrow}}{x_1 | x_2 \xrightarrow{\mathcal{I}_1 \mathcal{I}_2, (O_1^{\uparrow} O_2^{\uparrow})} \mathsf{v}} \qquad (jux3)$$

The three rules $(jux1)$, $(jux2)$ and $(jux3)$ deal with the cases in which δ occurs in the label of none, of the first and of both of the composed membrane systems, respectively. We omit the symmetric rule of $(jux2)$ in which δ occurs in the label of the second membrane system. We recall that, by definition of $(m3)$, $(m4)$ and $(fm2)$, we have that δ occurs in the label of the transition of a membrane only when such a membrane is dissolved (and transformed into v) by the application of some dissolving rule. The symbol δ is left in the label of the transition of the juxtaposition to denote dissolution of the whole system only when both of its components are dissolved.

To conclude with the semantics of the P Algebra we give the transition rules for hierarchical compositions of membranes.

$$\frac{x_1 \xrightarrow{M_1, \mathcal{I}_1^{\downarrow}, I_1^{\uparrow}, O_1^{\uparrow}, O_1^{\downarrow}} y_1 \qquad x_2 \xrightarrow{\mathcal{I}_2, O_2^{\uparrow}} y_2 \qquad \begin{array}{c} O_1^{\downarrow} \simeq \mathcal{I}_2 \quad O_2^{\uparrow} = I_1^{\uparrow} \quad \delta \notin O_1^{\uparrow} O_2^{\uparrow} \\ \nexists (l,a) \in M_1.(l \in \mathsf{Lab}(x_2) \wedge a \in \mathsf{Interface}(l, x_2)) \end{array}}{\mu(x_1, x_2) \xrightarrow{\mathcal{I}_1^{\downarrow}, O_1^{\uparrow}} \mu(y_1, y_2)} \qquad (h1)$$

$$\frac{x_1 \xrightarrow{M_1, \mathcal{I}_1^{\downarrow}, I_1^{\uparrow}, O_1^{\uparrow}, O_1^{\downarrow}} y_1 \qquad x_2 \xrightarrow{\mathcal{I}_2, O_2^{\uparrow}} y_2 \qquad \begin{array}{c} O_1^{\downarrow} \simeq \mathcal{I}_2 \quad O_2^{\uparrow} = I_1^{\uparrow} \quad \delta \in O_1^{\uparrow} \quad \delta \notin O_2^{\uparrow} \\ \nexists (l,a) \in M_1.(l \in \mathsf{Lab}(x_2) \wedge a \in \mathsf{Interface}(l, x_2)) \end{array}}{\mu(x_1, x_2) \xrightarrow{\mathcal{I}_1^{\downarrow}, O_1^{\uparrow} - \delta} y_2} \qquad (h2)$$

$$\frac{x_1 \xrightarrow{M_1, \mathcal{I}_1^{\downarrow}, I_1^{\uparrow}, O_1^{\uparrow}, O_1^{\downarrow}} y_1 \qquad x_2 \xrightarrow{\mathcal{I}_2, O_2^{\uparrow}} y_2 \qquad \begin{array}{c} O_1^{\downarrow} \simeq \mathcal{I}_2 \quad O_2^{\uparrow} - \delta = I_1^{\uparrow} \quad \delta \notin O_1^{\uparrow} \quad \delta \in O_2^{\uparrow} \\ \nexists (l,a) \in M_1.(l \in \mathsf{Lab}(x_2) \wedge a \in \mathsf{Interface}(l, x_2)) \end{array}}{\mu(x_1, x_2) \xrightarrow{\mathcal{I}_1^{\downarrow}, O_1^{\uparrow}} F(y_1)} \qquad (h3)$$

$$\frac{x_1 \xrightarrow{M_1, \mathcal{I}_1^{\downarrow}, I_1^{\uparrow}, O_1^{\uparrow}, O_1^{\downarrow}} y_1 \qquad x_2 \xrightarrow{\mathcal{I}_2, O_2^{\uparrow}} y_2 \qquad \begin{array}{c} O_1^{\downarrow} \simeq \mathcal{I}_2 \quad O_2^{\uparrow} - \delta = I_1^{\uparrow} \quad \delta \in O_1^{\uparrow} \cap O_2^{\uparrow} \\ \nexists (l,a) \in M_1.(l \in \mathsf{Lab}(x_2) \wedge a \in \mathsf{Interface}(l, x_2)) \end{array}}{\mu(x_1, x_2) \xrightarrow{\mathcal{I}_1^{\downarrow}, O_1^{\uparrow}} \mathsf{v}} \qquad (h4)$$

Rule $(h1)$ deals with the case in which neither the parent membrane nor the child membranes dissolve, rule $(h2)$ with the case in which the parent membrane dissolves, rule $(h3)$ with the case in which all the child membranes dissolve (and are transformed into v), and rule $(h4)$ with the case in which both the parent membrane and the child membranes dissolve. In all these transition rules it is checked that the objects sent from the parent membrane to child membranes and viceversa correspond to what is expected to be received from child membranes and the parent membrane, respectively. Moreover, in the premises of all these transition rules it is required that the assumption on the presence and on the interfaces of child membranes represented by M_1 are satisfied. Namely, if $(l, a) \in M_1$, then either no child membrane with label l is present or a does not appear in the interface of such a membrane. We assume $\doteqdot$ to be an equivalence relation on sets of pairs (l, u) with $l \in \mathbb{N}$ and $u \in V^*$, such that, given two such sets $\mathcal{I}_1$ and $\mathcal{I}_2$, then $\mathcal{I}_1 \doteqdot \mathcal{I}_2$ holds if and only if $(\mathcal{I}_1 \setminus \{(l, \varnothing) \mid l \in \mathbb{N}\}) = (\mathcal{I}_2 \setminus \{(l, \varnothing) \mid l \in \mathbb{N}\})$. We also assume $\mathsf{Lab}(x_1 \mid \ldots \mid x_n)$ and $\mathsf{Interface}(l, x_1 \mid \ldots \mid x_n)$ to be the set of labels of the juxtaposition of membranes, namely the set $\{l_1, \ldots, l_n\}$, and the interface of membrane $l_j = l$, respectively, where x_j is either $[\![_{l_j}\, \mathcal{R}, u\,]\!]^i_{l_j}$ or $\mu([_{l_j}\, \mathcal{R}, u\,]^i_{l_j}, x'_j)$.

All the semantics rules satisfy the constraints of the well–known *de Simone* format [8], which ensures that all the behavioural equivalences considered in [3] are congruences. In particular, we recall here the definition of bisimulation.

Definition 2.3. (Bisimulation)
Let $(\mathcal{S}, \mathcal{L}, \{\xrightarrow{\ell} \mid \ell \in \mathcal{L}\})$ be an LTS. A relation $R \subseteq \mathcal{S} \times \mathcal{S}$ is a *bisimulation* if for each pair of states (s_1, s_2) such that $s_1\, R\, s_2$ the following two conditions hold:

- $s_1 \xrightarrow{\ell} s'_1$ implies that there is a transition $s_2 \xrightarrow{\ell} s'_2$ such that $s'_1\, R\, s'_2$;
- $s_2 \xrightarrow{\ell} s'_2$ implies that there is a transition $s_1 \xrightarrow{\ell} s'_1$ such that $s'_1\, R\, s'_2$.

As usual, we denote with $\approx$ the union of all bisimulations (that is the largest bisimulation).

3. Some Axioms on Terms of the P Algebra

We give some basic axioms on the commutativity and associativity of the operations of the P Algebra, on some simple properties of evolution rules with promoters and inhibitors and on membranes with empty interfaces.

Definition 3.1. (Basic axioms)
Some *basic axioms* on terms of the P Algebra are the following:

$$c_1 \cup c_2 = c_2 \cup c_1 \qquad (\cup_1)$$
$$c_1 \cup (c_2 \cup c_3) = (c_1 \cup c_2) \cup c_3 \qquad (\cup_2)$$

$$ms_1 \mid ms_2 = ms_2 \mid ms_1 \qquad (|_1)$$
$$ms_1 \mid (ms_2 \mid ms_3) = ms_1 \mid (ms_2 \mid ms_3) \qquad (|_2)$$

$$(u \to v_h v_o \{v_{l_i}\}|_{Da\neg a}, \varnothing) = (\varnothing, \varnothing) \qquad (prom1)$$
$$(u \to v_h v_o \{v_{l_i}\}\delta|_{Da\neg a}, \varnothing) = (\varnothing, \varnothing) \qquad (prom2)$$
$$[_l\, au \to v_h v_o \{v_{l_i}\}|_{Da}\, ,\ \mathcal{R}\, ,\ u'\,]^i_l = [_l\, au \to v_h v_o \{v_{l_i}\}|_{D}\, ,\ \mathcal{R}\, ,\ u'\,]^i_l \qquad (prom3)$$

$$[_l\, au \to v_h v_o\{v_{l_i}\}\delta|_{Da}\,,\, \mathcal{R}\,,\, u'\,]_l^i = [_l\, au \to v_h v_o\{v_{l_i}\}\delta|_D\,,\, \mathcal{R}\,,\, u'\,]_l^i \quad (prom4)$$

$$[_l\, u \to v_h v_o\{v_{l_i}\}|_{Da}\,,\, u \to v_h v_o\{v_{l_i}\}|_{D\neg a}\,,\, \mathcal{R}\,,\, u'\,]_l^i = [_l\, u \to v_h v_o\{v_{l_i}\}|_D\,,\, \mathcal{R}\,,\, u'\,]_l^i \quad (prom5)$$

$$[_l\, u \to v_h v_o\{v_{l_i}\}\delta|_{Da}\,,\, u \to v_h v_o\{v_{l_i}\}\delta|_{D\neg a}\,,\, \mathcal{R}\,,\, u'\,]_l^i = [_l\, u \to v_h v_o\{v_{l_i}\}\delta|_D\,,\, \mathcal{R}\,,\, u'\,]_l^i \quad (prom6)$$

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

$$[\![_l\, \mathcal{R}, \varnothing\,]\!]_l^{\varnothing} = [\![_l\, u\,]\!]_l^{\varnothing} = [\![_l\, \varnothing\,]\!]_l^{\varnothing} \quad (if2)$$

$$ms = ms \mid [\![_l\, \varnothing\,]\!]_l^{\varnothing} \quad (if3)$$

$$[\![_{l_1}\, c\,]\!]_{l_1}^i = \mu\big([_{l_1}\, c\,]_{l_1}^i\,,\, [\![_{l_2}\, \varnothing\,]\!]_{l_2}^{\varnothing}\big) \quad (if4)$$

$$\frac{v_{l_i} \neq \varnothing}{[\![_l\, u \to v_h v_o\{v_{l_i}\}|_D\,,\, \mathcal{R}\,,\, u'\,]\!]_l^i = [\![_l\, \mathcal{R}\,,\, u'\,]\!]_l^i} \quad (fm1)$$

$$\frac{v_{l_i} \neq \varnothing}{[\![_l\, u \to v_h v_o\{v_{l_i}\}\delta|_D\,,\, \mathcal{R}\,,\, u'\,]\!]_l^i = [\![_l\, \mathcal{R}\,,\, u'\,]\!]_l^i} \quad (fm2)$$

The first four axioms state commutativity and associativity of union of membrane contents and juxtaposition of membrane systems.

Axioms $(prom1)$ and $(prom2)$ state that an evolution rule with an incoherent set of promoters and inhibitors can be removed from a membrane content. Axioms $(prom3)$ and $(prom4)$ state that an evolution rule having as a promoter one of the objects it consumes, can be simplified by removing such a promoter. These axioms equate two membranes containing $au \to v_h v_o\{v_{l_i}\}|_{Da}$ and $au \to v_h v_o\{v_{l_i}\}|_D$ rather than membrane contents $(au \to v_h v_o\{v_{l_i}\}|_{Da}, \varnothing)$ and $(au \to v_h v_o\{v_{l_i}\}|_D, \varnothing)$ because transitions performed by these two membrane contents have different sets of promoters as labels. Axioms $(prom5)$ and $(prom6)$ state that when a membrane contains two rules whose only difference is that one contains a promoter that is an inhibitor of the other, then such a membrane can be simplified by replacing these rules with one consisting of the common part of the two. Again, we define these axioms on membranes rather than membrane contents in order to ensure that the transitions performed by the equated terms are the same.

Axiom $(if1)$ states that if a membrane has empty interface, then its label l_1 can be changed into l_2. The reason is that the evolution rules of an outer membrane sending objects to l_1 are never applicable as the condition on the interface of l_1 is always violated. When the label is changed into l_2, then those evolution rules continue to be never applicable because they send objects to a membrane with label l_1 that is absent. On the other hand, no other rule can send objects to the new membrane l_2 because its interface is empty. The behaviour of a membrane with empty interface may consist only of objects sent out. Actually, if the membrane with empty interface is flat and contains either some rules but no object or some objects but no rule, then it is equivalent to an empty flat membrane. This is stated by axiom $(if2)$. Axioms $(if3)$ and $(if4)$ state that an empty flat membrane with an empty interface can be juxtaposed with any membrane system or inserted inside another membrane.

Axioms $(fm1)$ and $(fm2)$ state that in a flat membrane an evolution rule sending objects into inner membranes can be removed both in the case of a non–dissolving rule and in the case of a dissolving one.

Now we give some slightly more complex axioms.

Definition 3.2. (Axioms)
Some axioms on terms of the P Algebra are the following:

$$\frac{\nexists u' \rightarrow v'_h v'_o \{v_{l'_i}\}|_{D'} \in \mathcal{R} \text{ s.t. } a \in u'}{[_l\, \mathcal{R}\,,\, u \rightarrow v_h v_o \{v_{l_i}\}|_{D\neg a}\,,\, av\,]^i_l = [_l\, \mathcal{R}\,,\, av\,]^i_l} \qquad (ax1)$$

$$\frac{\nexists u' \rightarrow v'_h v'_o \{v_{l'_i}\}|_{D'} \in \mathcal{R} \text{ s.t. } a \in u'}{[_l\, \mathcal{R}\,,\, u \rightarrow v_h v_o \{v_{l_i}\}\delta|_{D\neg a}\,,\, av\,]^i_l = [_l\, \mathcal{R}\,,\, av\,]^i_l} \qquad (ax2)$$

$$\frac{a \notin u \qquad \nexists u' \rightarrow v'_h v'_o \{v_{l'_i}\}|_{D'} \in \mathcal{R} \text{ s.t. } a \in u'}{[_l\, \mathcal{R}\,,\, u \rightarrow v_h v_o \{v_{l_i}\}|_{Da}\,,\, av\,]^i_l = [_l\, \mathcal{R}\,,\, u \rightarrow v_h v_o \{v_{l_i}\}|_{D}\,,\, av\,]^i_l} \qquad (ax3)$$

$$\frac{a \notin u \qquad \nexists u' \rightarrow v'_h v'_o \{v_{l'_i}\}|_{D'} \in \mathcal{R} \text{ s.t. } a \in u'}{[_l\, \mathcal{R}\,,\, u \rightarrow v_h v_o \{v_{l_i}\}\delta|_{Da}\,,\, av\,]^i_l = [_l\, \mathcal{R}\,,\, u \rightarrow v_h v_o \{v_{l_i}\}\delta|_{D}\,,\, av\,]^i_l} \qquad (ax4)$$

$$\frac{a \notin v \qquad a \notin i \qquad \nexists u' \rightarrow v'_h v'_o \{v_{l'_i}\}|_{D'} \in \mathcal{R} \text{ s.t. } a \in v'_h}{[\![_l\, \mathcal{R}\,,\, u \rightarrow v_h v_o \{v_{l_i}\}|_{Da}\,,\, v\,]\!]^i_l = [\![_l\, \mathcal{R}\,,\, v\,]\!]^i_l} \qquad (ax5)$$

$$\frac{a \notin v \qquad a \notin i \qquad \nexists u' \rightarrow v'_h v'_o \{v_{l'_i}\}|_{D'} \in \mathcal{R} \text{ s.t. } a \in v'_h}{[\![_l\, \mathcal{R}\,,\, u \rightarrow v_h v_o \{v_{l_i}\}\delta|_{Da}\,,\, v\,]\!]^i_l = [\![_l\, \mathcal{R}\,,\, v\,]\!]^i_l} \qquad (ax6)$$

$$\frac{a \notin v \qquad a \notin v_h \qquad a \notin i \qquad \nexists u' \rightarrow v'_h v'_o \{v_{l'_i}\}|_{D'} \in \mathcal{R} \text{ s.t. } a \in v'_h}{[\![_l\, \mathcal{R}\,,\, u \rightarrow v_h v_o \{v_{l_i}\}|_{D\neg a}\,,\, v\,]\!]^i_l = [\![_l\, \mathcal{R}\,,\, u \rightarrow v_h v_o \{v_{l_i}\}|_{D}\,,\, v\,]\!]^i_l} \qquad (ax7)$$

$$\frac{a \notin v \qquad a \notin v_h \qquad a \notin i \qquad \nexists u' \rightarrow v'_h v'_o \{v_{l'_i}\}|_{D'} \in \mathcal{R} \text{ s.t. } a \in v'_h}{[\![_l\, \mathcal{R}\,,\, u \rightarrow v_h v_o \{v_{l_i}\}\delta|_{D\neg a}\,,\, v\,]\!]^i_l = [\![_l\, \mathcal{R}\,,\, u \rightarrow v_h v_o \{v_{l_i}\}\delta|_{D}\,,\, v\,]\!]^i_l} \qquad (ax8)$$

$$\frac{\nexists u \rightarrow v_h v_o \{v_{l_i}\}\delta|_{D} \in \mathcal{R} \wedge \ \nexists u \rightarrow v_h v_o \{v_{l_i}\}|_{D} \in \mathcal{R} \text{ s.t. } (a \in u) \vee (a \in D) \vee (\neg a \in D)}{[\![_l\, \mathcal{R}\,,\, av\,]\!]^i_l = [\![_l\, \mathcal{R}\,,\, v\,]\!]^i_l} \qquad (ax9)$$

Axioms $(ax1)$ and $(ax2)$ state that a membrane containing an object a that is an inhibitor of an evolution rule (non–dissolving and dissolving, respectively) can be simplified by removing such an evolution rule if there are no other non–dissolving rules consuming a. This condition ensures that the object a will always be present in the membrane, and consequently that the evolution rule with a as an inhibitor will never be applied. Moreover, this condition does not forbid dissolving rules to consume a, because after the application of these rules all the evolution rules of the membrane are lost.

Axioms $(ax3)$ and $(ax4)$ state that if in a membrane containing an object a there is an evolution rule (non–dissolving and dissolving, respectively) promoted by a, then, if such an evolution rule as well as all the other rules does not consume a, then a can be removed from the set of promoters and inhibitors of the rule. In fact, in this case the applicability of the evolution rule promoted by a will not depend on a itself because such an object will always be available.

Axioms $(ax5)$ and $(ax6)$ state that if in a flat membrane there is an evolution rule (non–dissolving and dissolving, respectively) that is promoted by a and if a is not present, is not produced by any rule and cannot be received from the external membrane (as it is not mentioned in the interface of the membrane), then the evolution rule promoted by a can be removed as it will never become applicable. Under

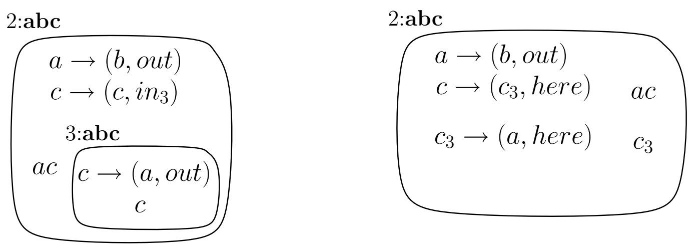

Figure 2. An example of flattening of a P System without dissolving rules.

similar conditions, axioms $(ax7)$ and $(ax8)$ state that an evolution rule (non–dissolving and dissolving, respectively) inhibited by a can be simplified by removing a from its inhibitors. In these four axioms the fact that the considered membrane is flat is important because otherwise some occurrences of a may become available by dissolution of inner membranes.

Finally, axiom $(ax9)$ states that if in a flat membrane there is an object a that is not consumed by any evolution rule and is neither a promoter nor an inhibitor of any rule, and if the membrane does not contain any dissolving rule, then the object a can be removed from the membrane.

4. Flattening Systems without Dissolving Rules

As shown in [4], 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. 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 $c_3 \to (a, here)$ which would result, after one more step, in the output of a b that would not be sent out by the original system. This problem is solved by a suitable use of membrane interfaces. In the example, interface abc of membrane 2 ensures that c_3 cannot be received from the environment.

Since the flattening technique we are going to define is based on renaming of objects, we need to describe how evolution rules are consequently changed. For this reason, we define two functions FlatIn and $\mathsf{FlatOut}$. 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 $\mathsf{FlatOut}$ we shall use the notation $u \to (v_h, here)(v_o, out)(v_{l_1}, in_{l_1}) \dots (v_{l_n}, in_{l_n})|_D$ for evolution rules rather than the more compact notation $u \to v_h v_o \{v_{l_i}\}|_D$.

We assume that the alphabet V is partitioned as follows: $V = V_\epsilon \cup \left(\bigcup_{L \in \mathbb{N}^+} V_L\right) \cup \{\hat{a}_l \mid a \in V_\epsilon, l \in \mathbb{N}\}$, where V_ϵ is the set of all objects without subscripts, V_L is the set of objects obtained by adding

$L \in \mathbb{N}^+$ as a subscript to each object of V_ϵ, and objects like $\hat{a}_l$ are special objects each obtained from an object $a \in V_\epsilon$ and a membrane label l. We denote with $\overline{V}$ and $\hat{V}$ the two sets of objects $\bigcup_{L\in\mathbb{N}^+} V_L$ and $\{\hat{a}_l \mid a \in V_\epsilon, l \in \mathbb{N}\}$, respectively, and it holds that $V_\epsilon, \overline{V}$ and $\hat{V}$ are pairwise disjoint. As an example, if $V_\epsilon = a, b, c, \ldots$, then $a_1, b_{1\cdot 2}$ and $c_{3\cdot 1\cdot 2}$ belong to $\overline{V}$, and $\hat{a}_2, \hat{b}_1$ and $\hat{c}_2$ belong to $\hat{V}$.

Let $\mathsf{Rid}(a_{L_1}, L_2)$ denote the object obtained by appending L_2 to the (possibly empty) subscript of an object $a_{L_1} \in V_\epsilon \cup \overline{V}$, namely $\mathsf{Rid}(a_{L_1}, L_2) = a_{L_1\cdot L_2}$. We assume that Rid does not change objects in $\hat{V}$, namely $\mathsf{Rid}(\hat{a}_l, L) = \hat{a}_l$. We extend the definition of Rid to sets and multisets of objects and to sets of promoters and inhibitors. For example, $\mathsf{Rid}(aabccc, 3) = a_3a_3b_3c_3c_3c_3 = a_3^2b_3c_3^3$, $\mathsf{Rid}(aa_{1\cdot 2}bb_2, 3\cdot 4) = a_{3\cdot 4}a_{1\cdot 2\cdot 3\cdot 4}b_{3\cdot 4}b_{2\cdot 3\cdot 4}$, and $\mathsf{Rid}(ab_1\neg c\neg d_2, 3) = a_3b_{1\cdot 3}\neg c_3\neg d_{2\cdot 3}$. Moreover, given a set of objects $S \subseteq V_\epsilon$, let $\mathsf{Hat}(S, l)$ denote the set $\{\hat{a}_l \mid a \in S\} \subseteq \hat{V}$. For example, $\mathsf{Hat}(\{a, b, c\}, 2) = \{\hat{a}_2, \hat{b}_2, \hat{c}_2\}$. The functions FlatIn and FlatOut are defined as follows:

$$\mathsf{FlatIn}\,(u \to (v_h, here)(v_o, out)|_D\ ,\ l) = \mathsf{Rid}(u, l) \to (\mathsf{Rid}(v_h, l)v_o, here)(\varnothing, out))|_{\mathsf{Rid}(D,l)}$$

$$\begin{aligned}\mathsf{FlatOut}\,(u \to (v_h, here)(v_o, out)(v_{l_1}, in_{l_1}) \ldots (v_{l_i}, in_{l_i}) \ldots (v_{l_n}, in_{l_n})|_D\ ,\ l_i) = \\ u \to (v_h\mathsf{Rid}(v_{l_i}, l_i), here)(v_o, out)(v_{l_1}, in_{l_1}) \ldots (\varnothing, in_{l_i}) \ldots (v_{l_n}, in_{l_n})|_{D\{\hat{a}_{l_i}|a\in v_{l_i}\}}\end{aligned}$$

Both FlatIn and FlatOut take a rule and a membrane label as arguments, and give a new rule as a 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.

Function FlatIn renames objects of the rules to which it is applied, by adding a subscript corresponding to the flattened membrane. Notice that, in the right part of the rule, only the objects that were originally sent to the membrane itself are renamed. The objects that were sent out are not renamed, simulating the fact that, after the application of the rule, they are no longer belonging to the flattened membrane. Moreover, notice that the rules to which FlatIn is applied do not send objects to any child membrane. This because we assume that the function FlatIn will be applied to rules belonging to flat membranes, and rules of a flat membrane sending objects to inner membranes can be removed by applying axioms $(fm1)$ and $(fm2)$.

Function FlatOut is applied to the rules of the membrane containing the flattened one. FlatOut replaces in each rule the objects sent to the flattened membrane l_i with objects to which subscript l_i is added. It is important to remark the role of the promoters $\hat{a}_{l_i}$ which are added to the rules given by FlatOut. Such promoters aim to simulate the interface of the flattened inner membrane. A rule is able to produce an object $a_{L\cdot l_i}$ (simulating an object a sent into inner membrane l_i) only if the promoter $\hat{a}_{l_i}$ is present. As we will show in the following, $\hat{a}_{l_i}$ is present only if the interface of the flattened membrane l_i contains a.

With abuse of notation we shall write $\mathsf{FlatIn}(\mathcal{R}, l)$ for $\{\mathsf{FlatIn}(r, l) \mid r \in \mathcal{R}\}$, and $\mathsf{FlatOut}(\mathcal{R}, l)$ for $\{\mathsf{FlatOut}(r, l) \mid r \in \mathcal{R}\}$. Now, the flattening technique is expressed by means of the following axioms.

Definition 4.1. (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

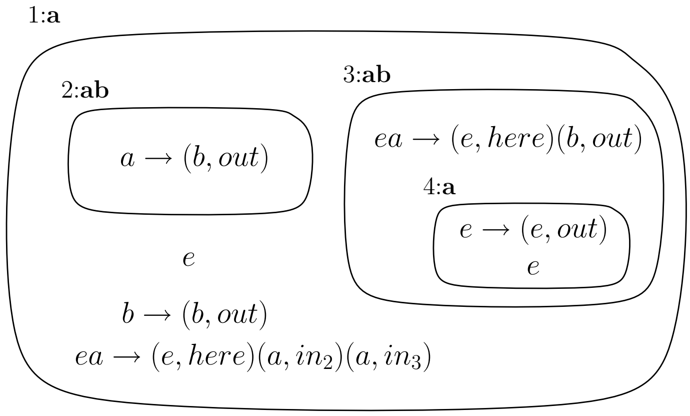

Figure 3. An example of P System.

objects in $V_{L\cdot l_2}$ for any $L \in \mathbb{N}^*$. The *flattening axioms* are the following:

$$\frac{ms \neq \mathsf{v} \quad L \in \mathbb{N}^* \quad V_{L\cdot l_2} \cap i_1 = \varnothing \quad \mathsf{FlatOut}(\mathcal{R}_1, l_2) = \mathcal{R}_1' \quad \mathsf{FlatIn}(\mathcal{R}_2, l_2) = \mathcal{R}_2'}{\mu\big([_{l_1}\, \mathcal{R}_1, u_1\,]_{l_1}^{i_1}\,,\, [\![_{l_2}\, \mathcal{R}_2, u_2\,]\!]_{l_2}^{i_2} \mid ms\big) = \mu\big([_{l_1}\, \mathcal{R}_1'\mathcal{R}_2', u_1\mathsf{Hat}(i_2, l_2)\mathsf{Rid}(u_2, l_2)\,]_{l_1}^{i_1}\,,\, ms\big)} \qquad (f1)$$

$$\frac{L \in \mathbb{N}^* \quad V_{L\cdot l_2} \cap i_1 = \varnothing \quad \mathsf{FlatOut}(\mathcal{R}_1, l_2) = \mathcal{R}_1' \quad \mathsf{FlatIn}(\mathcal{R}_2, l_2) = \mathcal{R}_2'}{\mu\big([_{l_1}\, \mathcal{R}_1, u_1\,]_{l_1}^{i_1}\,,\, [\![_{l_2}\, \mathcal{R}_2, u_2\,]\!]_{l_2}^{i_2}\big) = [\![_{l_1}\, \mathcal{R}_1'\mathcal{R}_2'\,,\, u_1\mathsf{Hat}(i_2, l_2)\mathsf{Rid}(u_2, l_2)\,]\!]_{l_1}^{i_1}} \qquad (f2)$$

Axioms $(f1)$ and $(f2)$ define the flattening of membranes. Axiom $(f2)$ is applicable to systems with a unique flat inner membrane, and axiom $(f1)$ is applicable when inner membranes are more than one. Both axioms apply FlatIn to the rules of the membrane to be flattened and FlatOut to the rules of the containing membrane. Notice that all the objects in the flattened membrane are renamed. Moreover, a set of special objects $\mathsf{Hat}(i_2)$ corresponding to the objects in the interface of the flattened membrane are produced. Such special objects might be used as promoters in the rules given by the function FlatOut.

As an example, let us consider the P System in Figure 3 which corresponds to the P Algebra term $t = \mu(m_1\,,\, m_2 \mid \mu(m_3, m_4))$ where:

$$\begin{aligned}
m_1 &= [_1\, \mathcal{R}_1, e\,]_1^a \text{ with } \mathcal{R}_1 = \{ea \to (e, here)(a, in_2)(a, in_3)\,,\, b \to (b, out)\}\\
m_2 &= [\![_2\, \mathcal{R}_2, \varnothing\,]\!]_2^{ab} \text{ with } \mathcal{R}_2 = \{a \to (b, out)\}\\
m_3 &= [_3\, \mathcal{R}_3, \varnothing\,]_3^{ab} \text{ with } \mathcal{R}_3 = \{ea \to (e, here)(b, out)\}\\
m_4 &= [\![_4\, \mathcal{R}_4, e\,]\!]_4^{a} \text{ with } \mathcal{R}_4 = \{e \to (e, out)\}\ .
\end{aligned}$$

Now, we have that

$$\begin{aligned}
\mu(m_3, m_4) &\overset{(f2)}{=} [\![_3\, \mathsf{FlatOut}(\mathcal{R}_3, 4)\mathsf{FlatIn}(\mathcal{R}_4, 4)\,,\, \varnothing\mathsf{Hat}(a, 4)\mathsf{Rid}(e, 4)\,]\!]_3^{ab}\\
&= [\![_3\, ea \to (e, here)(b, out)\,,\, e_4 \to (e, here)\,,\, \hat{a}_4 e_4\,]\!]_3^{ab}\ .
\end{aligned}$$

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

$$
\begin{aligned}
t &= \mu(m_1\,,\, m_2 \mid \mu(m_3, m_4)) = \mu(m_1\,,\, m_2 \mid fm_3) \\
&\stackrel{(f1)}{=} \mu(\,[_1\, \mathsf{FlatOut}(\mathcal{R}_1, 2)\mathsf{FlatIn}(\mathcal{R}_2, 2)\,,\, e\mathsf{Hat}(ab, 2)\mathsf{Rid}(\varnothing, 2)\,]_1^a\,, \\
&\qquad [\![_3\, ea \to (e, here)(b, out)\,,\, e_4 \to (e, here)\,,\, \hat{a}_4 e_4\,]\!]_3^{ab}\,) \\
&= \mu(\,[_1\, ea \to (ea_2, here)(a, in_3)|_{\hat{a}_2}\,,\, b \to (b, out)\,,\, a_2 \to (b, here)\,,\, \hat{a}_2\hat{b}_2 e\,]_1^a\,, \\
&\qquad [\![_3\, ea \to (e, here)(b, out)\,,\, e_4 \to (e, here)\,,\, \hat{a}_4 e_4\,]\!]_3^{ab}\,) \\
&\stackrel{(f2)}{=} [\![_1\, ea \to (ea_2 a_3, here)|_{\hat{a}_2\hat{a}_3}\,,\, b \to (b, out)\,,\, a_2 \to (b, here)\,, \\
&\qquad e_3 a_3 \to (e_3 b, here)\,,\, e_{43} \to (e_3, here)\,,\, \hat{a}_2\hat{b}_2\hat{a}_3\hat{b}_3\hat{a}_4 e e_{43}\,]\!]_1^a\,.
\end{aligned}
$$

Note that in the P Systems in Figure 3, rule $ea \to (e, here)(a, in_2)(a, in_3)$ in membrane 1 is enabled because a belongs to the interfaces of both membranes 2 and 3. In the flat membrane we have obtained by applying the axioms, the corresponding rule $ea \to (ea_2a_3, here)|_{\hat{a}_2\hat{a}_3}$ is enabled as well because special objects $\hat{a}_2$ and $\hat{a}_3$ are present in the content of the membrane. If a would not belong to the interface of either membrane 2 or membrane 3, the corresponding special objects would be absent from the content of the flat membrane. In this case, neither the original rule nor the one given by the flattening would be enabled.

Proposition 4.1. (soundness)
The portions of the LTS that are rooted in terms equated by axioms $(f1)$ and $(f2)$ are isomorphic.

Proof:
Let us begin with axiom $(f1)$. We prove that the portion of the LTS rooted in the P algebra term $\mu\big([_{l_1}\, \mathcal{R}_1, u_1\,]_{l_1}^{i_1}\,,\, [\![_{l_2}\, \mathcal{R}_2, u_2\,]\!]_{l_2}^{i_2} \mid ms\big)$ is isomorphic to a part of the portion of the LTS rooted in the term $\mu\big([_{l_1}\, \mathcal{R}_1'\mathcal{R}_2', u_1\mathsf{Hat}(i_2, l_2)\mathsf{Rid}(u_2, l_2)\,]_{l_1}^{i_1}\,,\, ms\big)$. More precisely, we prove that, given any transition $\mu\big([_{l_1}\, \mathcal{R}_1, u_1\,]_{l_1}^{i_1}\,,\, [\![_{l_2}\, \mathcal{R}_2, u_2\,]\!]_{l_2}^{i_2} \mid ms\big) \xrightarrow{\ell} t$, for an arbitrary term t, then there is a transition $\mu\big([_{l_1}\, \mathcal{R}_1'\mathcal{R}_2', u_1\mathsf{Hat}(i_2, l_2)\mathsf{Rid}(u_2, l_2)\,]_{l_1}^{i_1}\,,\, ms\big) \xrightarrow{\ell} t'$ such that t and t' are equated by the same axiom $(f1)$.

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

$$[_{l_1}\, \mathcal{R}_1, u_1\,]_{l_1}^{i_1} \xrightarrow{M_1, \mathcal{I}_1^\downarrow, I_1^\uparrow, O_1^\uparrow, O_1^\downarrow} [_{l_1}\, \mathcal{R}_1, u_1'\,]_{l_1}^{i_1} \tag{1}$$

$$[\![_{l_2}\, \mathcal{R}_2, u_2\,]\!]_{l_2}^{i_2} \xrightarrow{\mathcal{I}_2^\downarrow, O_2^\uparrow} [\![_{l_2}\, \mathcal{R}_2, u_2'\,]\!]_{l_2}^{i_2} \tag{2}$$

$$ms \xrightarrow{\mathcal{I}^\downarrow, O^\uparrow} ms' \tag{3}$$

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

$$[\![_{l_2}\, \mathcal{R}_2, u_2\,]\!]_{l_2}^{i_2} \mid ms \xrightarrow{\mathcal{I}_2^\downarrow\mathcal{I}^\downarrow, O_2^\uparrow O^\uparrow} [\![_{l_2}\, \mathcal{R}_2, u_2'\,]\!]_{l_2}^{i_2} \mid ms' \tag{4}$$

through semantic rule $(jux1)$. The transition from $\mu\big([_{l_1}\, \mathcal{R}_1, u_1\,]^{i_1}_{l_1}, [\![_{l_2}\, \mathcal{R}_2, u_2\,]\!]^{i_2}_{l_2} \;\mid\; ms\big)$ is inferred through semantic rule $(h1)$ from (1) and (4) and takes the shape:

$$\mu\big([_{l_1}\, \mathcal{R}_1, u_1\,]^{i_1}_{l_1}\,,\; [\![_{l_2}\, \mathcal{R}_2, u_2\,]\!]^{i_2}_{l_2} \mid ms\big) \xrightarrow{\mathcal{I}_1^{\downarrow}, O_1^{\uparrow}} \mu\big([_{l_1}\, \mathcal{R}_1, u_1'\,]^{i_1}_{l_1}\,,\; [\![_{l_2}\, \mathcal{R}_2, u_2'\,]\!]^{i_2}_{l_2} \mid ms'\big) \tag{5}$$

where:

1. $O_1^{\downarrow} \simeq \mathcal{I}_2^{\downarrow}\mathcal{I}^{\downarrow}$
2. $O^{\uparrow}O_2^{\uparrow} = I_1^{\uparrow}$
3. $\nexists(l,a) \in M_1.\, (l = l_2 \wedge a \in i_2) \vee (l \in \mathsf{Labels}(ms) \wedge a \in \mathsf{Interface}(l, ms))$.

Transition (2) is inferred from

$$[_{l_2}\, \mathcal{R}_2, u_2\,]^{i_2}_{l_2} \xrightarrow{M_2, \mathcal{I}_2^{\downarrow}, \varnothing, O_2^{\uparrow}, \varnothing} [_{l_2}\, \mathcal{R}_2, u_2'\,]^{i_2}_{l_2} \tag{6}$$

through the semantic rules $(fm1)$, for some M_2.

Now, (1) can be inferred through $(m2)$ or $(m1)$. In the former case $\mathcal{I}_1^{\downarrow} = \{(l_1, I_1)\}$, for some I_1, in the latter case $\mathcal{I}_1^{\downarrow} = \varnothing$. Analogously, (6) can be inferred through $(m2)$ or $(m1)$. In the former case $\mathcal{I}_2^{\downarrow} = \{(l_2, I_2)\}$, for some I_2 such that $\mathsf{Set}(I_2) \subseteq i_2$, in the latter case $\mathcal{I}_2^{\downarrow} = \varnothing$. We assume that both (1) and (6) are inferred from $(m2)$, the other cases are similar. The transitions originating (1) and (6) have the shape:

$$(\mathcal{R}_1, u_1) \xrightarrow[v_1, U_1, v_1, v_1', D_1]{M_1, I_1 I_1^{\uparrow}, O_1^{\uparrow}, O_1^{\downarrow}} (\mathcal{R}_1, u_1') \tag{7}$$

$$(\mathcal{R}_2, u_2) \xrightarrow[v_2, U_2, v_2, v_2', D_2]{M_2, I_2, O_2^{\uparrow}, \varnothing} (\mathcal{R}_2, u_2') \tag{8}$$

for suitable values $v_1, U_1, v_1', D_1, v_2, U_2, v_2', D_2$ with $D_1^+ = D_2^+ = \varnothing$.

Notice that this implies that $\mathsf{Set}(I_1) \subseteq i_1$. Note also that $O_1^{\downarrow} \simeq \mathcal{I}_2\mathcal{I}$ implies that there exists some $O_1'^{\downarrow}$ such that

$$O_1^{\downarrow} = \{(l_2, I_2)\} \cup O_1'^{\downarrow} \tag{9}$$

From (7) we infer

$$(\mathcal{R}_1', u_1) \xrightarrow[v_1, U_1, v_1, v_1', D_1\{\hat{a}_{l_2} \mid a \in i_2\}]{M_1, I_1 I_1^{\uparrow}, O_1^{\uparrow}, O_1'^{\downarrow}} (\mathcal{R}_1', u_1' \mathsf{Rid}(I_2, l_2)) \tag{10}$$

By removing input $O_2^{\uparrow}$ from the input of (10), the same transition fires but $O_2^{\uparrow}$ does not appear in u_1':

$$(\mathcal{R}_1', u_1) \xrightarrow[v_1, U_1, v_1, v_1', D_1\{\hat{a}_{l_2} \mid a \in i_2\}]{M_1, I_1 I_1^{\uparrow} \setminus O_2^{\uparrow}, O_1^{\uparrow}, O_1'^{\downarrow}} (\mathcal{R}_1', (u_1' \mathsf{Rid}(I_2, l_2)) \setminus O_2^{\uparrow}) \tag{11}$$

From (8) we infer:

$$(\mathcal{R}_2', \mathsf{Hat}(i_2, l_2)\mathsf{Rid}(u_2, l_2)) \xrightarrow[\overline{v_2}, \overline{U_2}, \overline{v_2}, \overline{v_2'}, \mathsf{Rid}(D_2, l_2)]{M_2, \varnothing, \varnothing, \varnothing} (\mathcal{R}_2', \mathsf{Hat}(i_2, l_2)\mathsf{Rid}(u_2' \setminus I_2, l_2) O_2^{\uparrow}) \tag{12}$$

where $\overline{v_2}, \overline{U_2}, \overline{v_2'}$ denote $\mathsf{Rid}(v_2, l_2)$, $\mathsf{Rid}(U_2, l_2)$, $\mathsf{Rid}(v_2', l_2) \cup \{\hat{a}_{l_2} \mid a \in i_2\}$, respectively.
Through semantic rule $(u1)$, from (11) and (12) we infer

$$(\mathcal{R}_1'\mathcal{R}_2', u_1\mathsf{Hat}(i_2, l_2)\mathsf{Rid}(u_2, l_2)) \xrightarrow[v_1\overline{v_2}, U_1 \oplus \overline{U_2}, v_1\overline{v_2}, v_1'\overline{v_2'}, D_1\mathsf{Rid}(D_2, l_2)]{M_1M_2, I_1 I_1^{\uparrow} \setminus O_2^{\uparrow}, O_1^{\uparrow}, O_1'^{\downarrow}} (\mathcal{R}_1'\mathcal{R}_2', u_1'\mathsf{Hat}(i_2, l_2)\mathsf{Rid}(u_2', l_2)) \tag{13}$$

Notice that all objects $\hat{a}_{l_2}$ such that $a \in i_2$ are not promoters of this transition. They have been removed since they appear in $\overline{v_2'}$.
By applying semantic rule $(m2)$, which is applicable since $\mathsf{Set}(I_1) \subseteq i_1$, we infer:

$$[_{l_1}\, \mathcal{R}_1'\mathcal{R}_2', u_1\mathsf{Hat}(i_2, l_2)\mathsf{Rid}(u_2, l_2)\,]_{l_1}^{i_1} \xrightarrow{M_1M_2, \{(l_1, I_1)\}, I_1^{\uparrow} \setminus O_2^{\uparrow}, O_1^{\uparrow}, O_1'^{\downarrow}} [_{l_1}\, \mathcal{R}_1'\mathcal{R}_2', u_1'\mathsf{Hat}(i_2, l_2)\mathsf{Rid}(u_2', l_2)\,]_{l_1}^{i_1} \tag{14}$$

We already know that $O_1^{\downarrow} \doteq \mathcal{I} \cup \{(l_2, I_2)\}$, $O^{\uparrow}O_2^{\uparrow} = I_1^{\uparrow}$, $O_1^{\downarrow} = \{(l_2, I_2)\} \cup O_1'^{\downarrow}$ and $\nexists (l, a) \in M_1.\, (l = l_2 \wedge a \in i_2) \vee (l \in \mathsf{Labels}(ms) \wedge a \in \mathsf{Interface}(l, ms))$. Therefore, $O_1'^{\downarrow} \doteq \mathcal{I}$, $O^{\uparrow} = I_1^{\uparrow} \setminus O_2^{\uparrow}$ and $\nexists (l, a) \in M_1.\, (l \in \mathsf{Labels}(ms) \wedge a \in \mathsf{Interface}(l, ms))$. So, we can apply the semantic rule $(h1)$ to infer that (3) and (14) originate

$$\mu\big([_{l_1}\, \mathcal{R}_1'\mathcal{R}_2', u_1\mathsf{Hat}(i_2, l_2)\mathsf{Rid}(u_2, l_2)\,]_{l_1}^{i_1}\,,\, ms\big) \xrightarrow{\mathcal{I}_1^{\downarrow}, O_1^{\uparrow}} \mu\big([_{l_1}\, \mathcal{R}_1'\mathcal{R}_2', u_1'\mathsf{Hat}(i_2, l_2)\mathsf{Rid}(u_2', l_2)\,]_{l_1}^{i_1}\,,\, ms'\big) \tag{15}$$

Summarizing, given any transition from $\mu\big([_{l_1}\, \mathcal{R}_1, u_1\,]_{l_1}^{i_1}\,,\, [\![_{l_2}\, \mathcal{R}_2, u_2\,]\!]_{l_2}^{i_2} \mid ms\big)$, we have a corresponding transition from $\mu\big([_{l_1}\, \mathcal{R}_1'\mathcal{R}_2', u_1\mathsf{Hat}(i_2, l_2)\mathsf{Rid}(u_2, l_2)\,]_{l_1}^{i_1}\,,\, ms\big)$, where the two transitions have the same label and take to terms related by axiom $(f1)$. The converse is similar, with the use of premise $V_{L \cdot l_2} \cap i_1 = \varnothing$.

The proof of the case of axiom $(f2)$ is analogous to that of $(f1)$, since basic axioms allow us to rewrite $[\![_{l_1}\, \mathcal{R}_1'\mathcal{R}_2'\,,\, u_1\mathsf{Hat}(i_2, l_2)\mathsf{Rid}(u_2, l_2)\,]\!]_{l_1}^{i_1}$ into $\mu\big([_{l_1}\, \mathcal{R}_1'\mathcal{R}_2'\,,\, u_1\mathsf{Hat}(i_2, l_2)\mathsf{Rid}(u_2, l_2)\,]_{l_1}^{i_1}\,,\, [\![_{l}\, \varnothing\,]\!]_{l}^{\varnothing}\big)$, and $\mu\big([_{l_1}\, \mathcal{R}_1, u_1\,]_{l_1}^{i_1}\,,\, [\![_{l_2}\, \mathcal{R}_2, u_2\,]\!]_{l_2}^{i_2}\big)$ into $\mu\big([_{l_1}\, \mathcal{R}_1, u_1\,]_{l_1}^{i_1}\,,\, [\![_{l_2}\, \mathcal{R}_2, u_2\,]\!]_{l_2}^{i_2} \mid [\![_{l}\, \varnothing\,]\!]_{l}^{\varnothing}\big)$. □

Theorem 4.1. Any membrane system $\mu(m, ms)$ with multisets of objects and evolution rules built over V_ϵ and without dissolving rules, can be reduced to a flat membrane $[\![_l\, c\,]\!]_l^i$ such that $\mu(m, ms) \approx [\![_l\, c\,]\!]_l^i$.

Proof:
This can be proved by induction on the number of membrane nodes in the membrane nesting tree of $\mu(m, ms)$. If such a tree contains two membranes, namely $\mu(m, ms)$ is $\mu([_{l_1}\, c_1\,]_{l_1}^{i_1}, [\![_{l_2}\, c_2\,]\!]_{l_2}^{i_2})$, then the proof follows immediately from axiom $(f2)$. If the membrane nesting tree contains more that two membranes, then the proof can be done by resorting to the induction hypothesis after applying one of the axioms $(f1)$ 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 V_ϵ, ensure that the assumptions and the premises of the axioms are always satisfied. Finally, Proposition 4.1 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. □

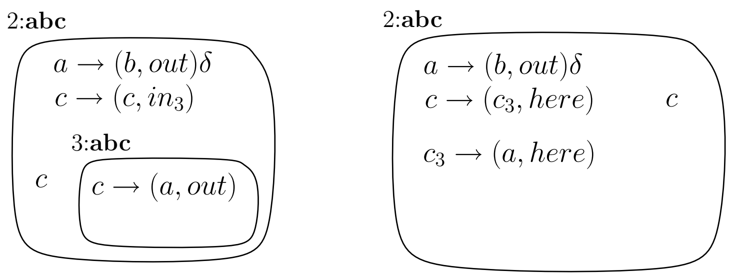

Figure 4. Example of application of the flattening technique defined in Section 4 to a P System containing a dissolving rule.

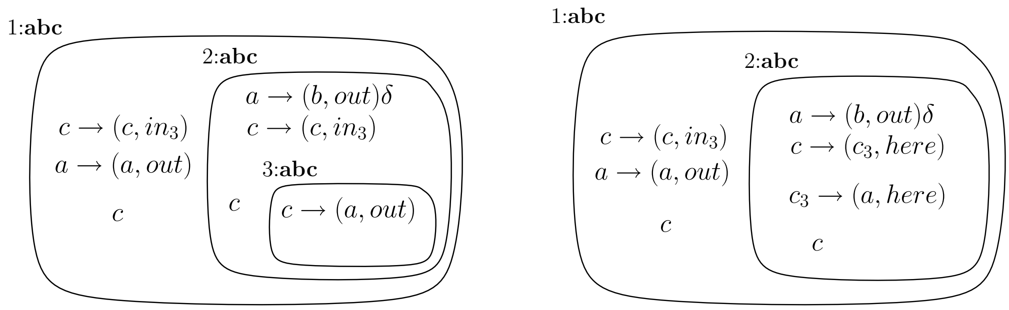

Figure 5. Example of context in which the two membrane systems of Figure 4 behave differently.

5. Flattening Systems with Dissolving Rules

The flattening technique explained in the previous section cannot be applied if membranes contain dissolving rules. As an example, let us consider Figure 4, where flattening is applied, as defined in the previous section, to a P System containing a dissolving rule. When membrane 2 does not receive any object from the environment, both the original P System and the corresponding flat system send object b to the external environment after two steps and dissolve. The difference between the two cases is that dissolution of membrane 2 in the original P System leads to a configuration in which membrane 3 is the outmost (and only) membrane, while dissolution of membrane 2 in the flat system leads to configuration in which the whole system disappears. As a consequence, we can provide a context in which the original membrane system and the flat membrane behave differently (see Figure 5). The point 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 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) the rules of such a membrane disappear, and (iii) the 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

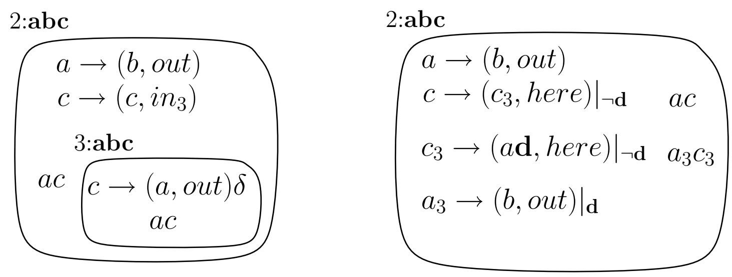

Figure 6. Example of flattening in which membrane dissolution is simulated.

flattening. This would allow (ii) and (iii) to be simulated by using $\mathbf{d}$ as an inhibitor of the rules of the dissolved membrane and of the rules sending objects to the dissolved membrane. Objects of the dissolved membrane can be renamed. Correspondingly, rules of the outer membranes are duplicated and changed in such a way that they can apply to the renamed objects. The new rules have $\mathbf{d}$ has a promoter.

We give a simple example of flattening with dissolution of inner membranes in Figure 6. 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.

Let us assume the set of objects V to be partitioned as follows: $V = V_\epsilon \cup \overline{V} \cup \hat{V} \cup \{\mathbf{d}_i \mid i \in \mathbb{N}\} \cup \{\mathbf{s}_L \mid L \in \mathbb{N}^+\}$, where $V_\epsilon, \overline{V}$ and $\hat{V}$ are as in the previous section, object $\mathbf{d}_i$ indicates that membrane i has been dissolved, and object $\mathbf{s}_{l_1 \cdot \ldots \cdot l_n}$ in membrane l indicates that l_1 is a child of l_2, $\ldots$, l_{n-1} is a child of l_n, which, in turn, is a child of l. We denote the sets of objects $\{\mathbf{d}_i \mid i \in \mathbb{N}\}$ and $\{\mathbf{s}_L \mid L \in \mathbb{N}^+\}$ with $\mathbf{D}$ and $\mathbf{S}$, respectively, and we extend the definition of Rid given in the previous section in such a way that it changes objects in $\mathbf{S}$ and it does not change objects in $\mathbf{D}$. In other words, $\mathsf{Rid}(\mathbf{s}_{L_1}, L_2) = \mathbf{s}_{L_1 \cdot L_2}$ and $\mathsf{Rid}(\mathbf{d}_l, L) = \mathbf{d}_l$.

Let us define functions FlatIn' and $\mathsf{FlatOut}'$, having the same role of functions FlatIn and $\mathsf{FlatOut}$ of the previous section. Our aim is to flatten all child membranes of a given membrane l' at the same time, by applying FlatIn' to rules of these membranes and $\mathsf{FlatOut}'$ to rules of l'. Moreover, this flattening is done only if all child membranes of l' are flat.

Function FlatIn' has two arguments: a rule contained in the membrane l we want to flatten, and l itself. We have several cases, depending on the origin of the rule, namely membrane l or a membrane originally inside l and already flattened, and depending on whether the rule is dissolving or not.

Let us consider first the case in which a rule in membrane l originates from a rule in a membrane l'' inside l that has been already flattened. In this case the original rule was already modified by FlatIn' when l'' was flattened, and has the form $u \to (v_h, here)|_D$, where u has at least one object a_σ with $l'' \in \sigma$ and $\mathbf{d}_{l''} \notin D$. All the objects in the rule must be renamed, in order to take into account the structure of membranes. As the rule was not originally in l, it should be applicable independently from the dissolution of l. Hence, inhibitor $\mathbf{d}_l$ is not added to the rule and we have:

$$\mathsf{FlatIn}'\,(u \to (v_h, here)|_D\;,\; l) = \mathsf{Rid}(u,l) \to (\mathsf{Rid}(v_h,l), here)|_{\mathsf{Rid}(D,l)} \qquad \text{if } \exists a_\sigma \in u.\, \exists l'' \in \sigma.\, \mathbf{d}_{l''} \notin D\,.$$

If the rule corresponds to a rule originally in l, we have to distinguish whether the rule is dissolving or not. Let us begin with a non–dissolving rule $u \to (v_h, here)(v_o, out)|_D$. Note that the rule does not send objects to any child membrane, which is a condition satisfied by all the rules if l is flat. We rename with label l all the objects triggering the rule and all the objects produced by the rule and with l as a destination, so that they are not confused with the objects in the outer membrane. We do not rename objects that were sent out, so that if the new rule fires then the outer membrane has these objects available. The new rule requires that l has not been dissolved, which is expressed by the inhibitor $\mathbf{d}_l$. Hence, we have

$$\mathsf{FlatIn}'\,(u \to (v_h, here)(v_o, out)|_D)\,,\, l) = \mathsf{Rid}(u,l) \to (\mathsf{Rid}(v_h,l)v_o, here)|_{\mathsf{Rid}(D,l)\neg\mathbf{d}_l} \qquad \text{if } \forall a_\sigma \in u.\, (\sigma = \epsilon \vee \forall l' \in \sigma.\, \mathbf{d}_{l'} \in D)\,.$$

The last case deals with a dissolving rule, and requires two changes with respect to the previous case. First, the new rule produces object $\mathbf{d}_l$, expressing that membrane l has dissolved. Second, the objects that were originally produced by the rule and with l as a destination are not renamed, so that they are available in the outer membrane after the application of the rule. Hence, we have

$$\mathsf{FlatIn}'\,(u \to (v_h, here)(v_o, out)\delta|_D)\,,\, l) = \mathsf{Rid}(u,l) \to (v_h v_o \mathbf{d}_l, here)|_{\mathsf{Rid}(D,l)\neg\mathbf{d}_l} \qquad \text{if } \forall a_\sigma \in u.\, (\sigma = \epsilon \vee \forall l' \in \sigma.\, \mathbf{d}_{l'} \in D)\,.$$

Notice that when a rule r obtained through FlatIn' produces $\mathbf{d}_l$, then all rules originally in l and obtained through FlatIn' are correctly locked by the inhibitor $\mathbf{d}_l$. Given any non–dissolving rule r' of the form $u \to (v_h, here)(v_o, out)|_D$, all objects in v_h are marked with l by FlatIn'. These objects should be available in the outer membrane due to the dissolution of l caused by the firing of r. The problem will be solved by function $\mathsf{FlatOut}'$ applied to the outer membrane.

Note that we have not considered rules corresponding to rules of the form $u \to (v_h, here)\delta|_D$ originated by the flattening of inner membranes as FlatIn' applied to inner membranes replaces δ with $\mathbf{d}$.

In the following, given a set of rules $\mathcal{R}$ in a membrane l, we shall denote with $\mathcal{R}_l$ the subset in $\mathcal{R}$ of the rules originally in membrane l.

In order to define $\mathsf{FlatOut}'$ we need the following definition. Let $V_a = \{a_L \mid L \in \mathbb{N}^+\}$. Given a rule r of the form $a1 \ldots an \to v_h v_o \{v_{l_i}\}|_D$, and a set $O \subset \overline{V}$, we denote with $\mathsf{occ}(r, O)$ the set of the rules

$$b1 \ldots bn \to v_h v_o \{v_{l_i}\}|_{DD'}$$

such that:

- $bi \in \{ai\} \cup (O \cap V_{ai})$ for each $1 \le i \le n$;
- $D' = \{\mathbf{d}_l \mid bi = ai_{L_i}, l \in L_i, 1 \le i \le n\}$.

Assume that r is in a membrane l. Any rule in $\mathsf{occ}(r, \{V_{ai}\})$ requires an object $ai_{L_i} \in V_{ai}$, instead of ai, for all $1 \leq i \leq n$. If all membranes contained in l have been flattened and $L_i \neq \epsilon$, then object ai_{L_i} occurs in l if ai occurred originally inside nested membranes contained in l and having label in L_i. In the original system, if all these membranes have been dissolved then ai was available in l to apply r. Hence, if all promoters $\mathbf{d}_l$ with $l \in L_i$ are now in l, then ai_{L_i} can trigger the rule.

Let us define now function $\mathsf{FlatOut}'$. It has three arguments, namely a rule in a membrane l', a set of objects contained in $\mathbf{S}$ showing the path between l' and the inner membranes to which the rule sends objects, and a set of objects of the form a_σ that can be used to trigger the rule instead of a and that are inherited from child membranes that have dissolved.

$$\mathsf{FlatOut}'\big(u \to (v_h, here)(v_o, out)\{(v_{l_i}, in_{l_i}) \,|\, i \in I\}|_D \,,\, U \cup \bigcup_{i\in I} \mathbf{s}_{l_i\cdot\sigma_i}, O\big) =$$

$$\mathsf{occ}\big(u \to (v_h\{\mathsf{Rid}(v_{l_i}, l_i \cdot \sigma_i) \,|\, i \in I\}, here)(v_o, out)|_{D\cup\{\neg\mathbf{d}_{l_i}|i\in I\}\cup\{\mathbf{d}_l|l\in\bigcup_{i\in I}\sigma_i\}\cup\bigcup_{i\in I}\{\hat{a}_{l_i}|a\in v_{l_i}\}}, O\big)$$

Assume that we aim to flatten membranes inside l' to l'. In this case the second argument of $\mathsf{FlatOut}'$ shall be a set of objects describing the original structure of these membranes. Since the original rule sends objects to each membrane l_i with $i \in I$, the rule was originally applicable only when all l_i's are children of l'. Now, the objects $\mathbf{s}_{l_i\cdot\sigma_i}$ indicate that all membranes whose label is in σ_i were originally between l' and the target membrane l_i, hence the rule was originally applicable only after dissolutions of all these membranes. This is why we add promoters $\{\mathbf{d}_l \,|\, l \in \bigcup_{i\in I}\sigma_i\}$ to all rules obtained through $\mathsf{FlatOut}'$. Moreover, these rules have inhibitors $\{\mathbf{d}_{l_i} \,|\, i \in I\}$ since the original rule was applicable only if all l_i's were not dissolved. Finally, all objects in v_{l_i} remain now in l', after they have been renamed by adding the label $l_i \cdot \sigma_i$. The third argument of $\mathsf{FlatOut}'$ shall be a set of objects of the form $a_{\sigma\cdot l'_i}$ such that l'_i is one of the inner membranes and a_σ is in l'_i. By applying function occ we generate several rules, where each object a in u may be replaced by an object $a_{\sigma\cdot l'_i}$ such that a_σ is in an inner membrane l'_i. Function occ ensures that for each membrane label l in $\sigma \cdot l'_i$ the corresponding promoter $\mathbf{d}_l$ is added to the rule. In this way we are sure that objects originally in some membrane inside l'_i, or in l'_i itself, can be exploited by rules in l' after dissolution of l'_i and of all other membranes containing them.

Let us extend FlatIn' and $\mathsf{FlatOut}'$ to sets of rules:

$$\mathsf{FlatIn}'(\mathcal{R}, l) = \bigcup_{r\in R} \mathsf{FlatIn}'(r, l)$$

$$\mathsf{FlatOut}'(R, O, O') = \bigcup_{r\in R} \mathsf{FlatOut}'(r, O, O')$$

Given membrane systems $\{ms_1, \ldots, ms_n\}$ such that either $ms_i = \mu(m, ms'_i)$ or $ms_i = F(m)$, axioms $|_1$ and $|_2$ permit us to write $ms_1 \mid \cdots \mid ms_n$ to denote their juxtaposition, which will be sometimes denoted $\bigcup_{1\leq i\leq n} ms_i$.

For a set of rules $\mathcal{R}$, let $\mathsf{obj}(\mathcal{R})$ denote the set of the objects appearing in the rules.

Definition 5.1. (Flattening axiom)
Let $\mathcal{R}$ and $(\mathcal{R}_i)_{i\in I}$ be sets of evolution rules, and let $\mathcal{R}$ and u contain no objects in $V_{L\cdot l_i}$. The *flattening*

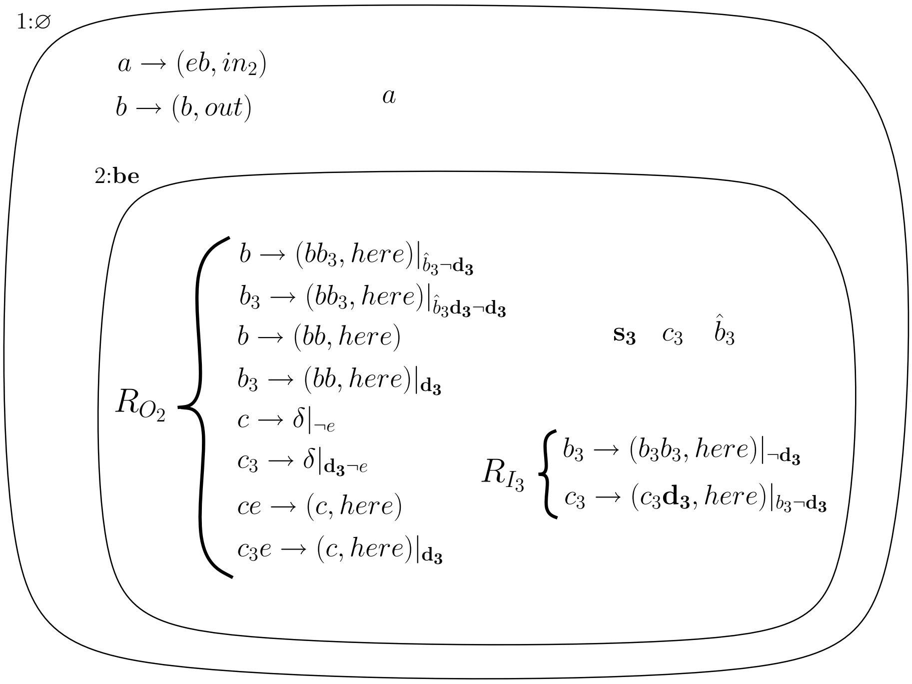

Figure 7. The result of the application of the flattening axiom to the P System of Figure 1

axiom is the following:

$$\frac{L \in \mathbb{N}^* \qquad (\bigcup_{j \in I} V_{L \cdot l_j}) \cap i = \varnothing \qquad \mathsf{FlatIn}'(\mathcal{R}_j, l_j) = \mathcal{R}'_j \\ \mathsf{FlatOut}'\left(\mathcal{R}, \bigcup_{j \in I}\{\mathbf{s}_{\sigma \cdot l_j} \mid \mathbf{s}_\sigma \in u_j\} \bigcup_{j \in I} \mathbf{s}_{l_j}, \bigcup_{j \in I} \mathsf{Set}(\mathsf{Rid}(u_j \mathsf{obj}(\mathcal{R}_j)), l_j)\right) = \mathcal{R}'}{\mu\left([_l\, \mathcal{R}, u\,]^i_l\,, \bigcup_{j \in I} [\![_{l_j}\, \mathcal{R}_j, u_j\,]\!]^{i_j}_{l_j}\right) = [\![_l\, \mathcal{R}' \cup \bigcup_{j \in I} \mathcal{R}'_j, u \bigcup_{j \in I} \mathsf{Hat}(i_j, l_j) \bigcup_{j \in I} \mathbf{s}_{l_j} \bigcup_{j \in I} \mathsf{Rid}(u_j, l_j)\,]\!]^i_l}$$

The flattening axiom permits us to flatten all flat membranes contained in l, into l itself. The objects in the second argument of $\mathsf{FlatOut}'$ give all needed information about the structure of l_j. All objects in u_j are renamed by adding the tag l_j, since they should not be confused with objects in l. The object $\mathbf{s}_{l_j}$ is added to the objects of l, to represent that l originally contained l_j. If l is flattened into another membrane l', then $\mathbf{s}_{l_j}$ will be renamed $\mathbf{s}_{l_j \cdot l}$, to represent the origin of membrane l_j.

Let us show an application of the described flattening technique. Let us consider again the P System in Figure 1. Membranes 2 and 3 contain dissolving rules. By applying the flattening axiom to membranes 2 and 3 we obtain the system in Figure 7, where the rules in the set R_{I_3} are obtained by applying function FlatIn' to the rules of membrane 3, and the rules in the set R_{O_2} are obtained by applying function $\mathsf{FlatOut}'$ to the rules in membrane 2. Note that the second rule in R_{O_2} is never applicable as its set of promoters and inhibitors is incoherent. This rule could be removed by applying axiom $(prom1)$. Finally, by applying the flattening axiom to membranes 1 and 2 in Figure 7 we obtain the flat system in Figure 8. Rules in R_{I_2} are obtained by applying function FlatIn' to membrane 2, and rules in R_{O_1} are obtained by applying function $\mathsf{FlatOut}'$ to membrane 1.

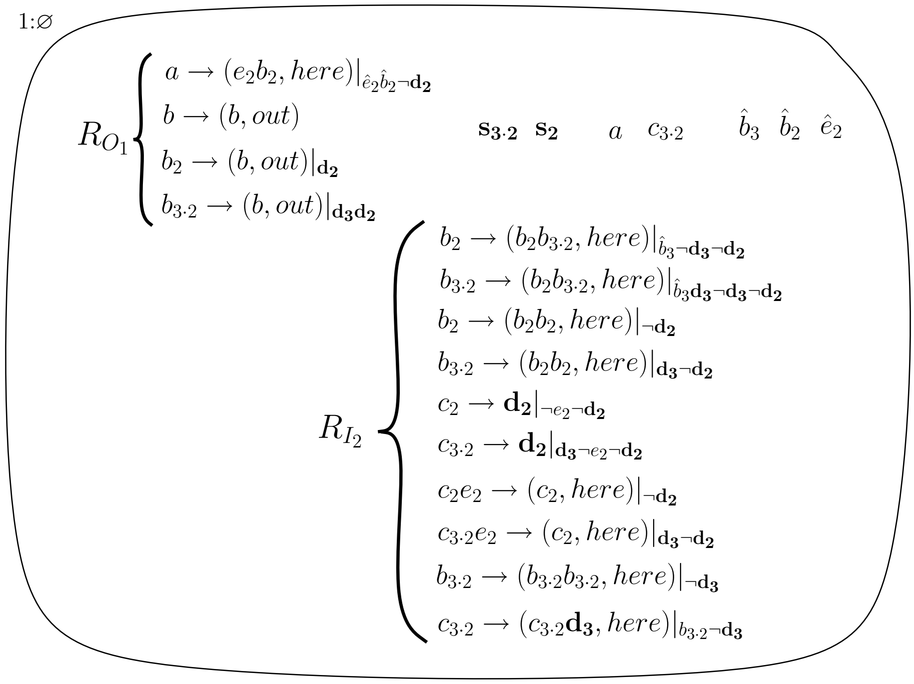

Figure 8. The result of the application of the flattening axiom to the P System of Figure 7

Theorem 5.1. (soundness)
Let us assume a membrane system ms and a flat membrane $[\![_l\, c\,]\!]_l^i$ such that $ms = [\![_l\, c\,]\!]_l^i$ is inferred by applying the flattening axiom repeatedly in the tree structure of ms. If ms cannot perform step in which its outmost membrane dissolve, then $ms \approx [\![_l\, c\,]\!]_l^i$.

Proof:
The theorem follows from property (16) in Lemma 5.1. □

Given a set of objects v and a rule r, we write $v \not\vdash r$ if v contains an object that is an inhibitor of r.

If a membrane system ms is not a juxtaposition of other membrane systems, let $\mathsf{root}(ms)$ denote the label of the outmost membrane in ms, if any. Formally, $\mathsf{root}(ms) = l$ if either $ms = \mu([_l\, c\,]_l^i, ms')$ or $ms = [\![_l\, c\,]\!]_l^i$, and $\mathsf{root}(ms) = \perp$, otherwise.

Lemma 5.1. Let us assume a membrane system ms and a flat membrane $[\![_l\, \mathcal{R}, u\,]\!]_l^i$ such that $ms = [\![_l\, \mathcal{R}, u\,]\!]_l^i$ is inferred by applying the flattening axiom repeatedly in the tree structure of ms. It holds that:

$$ms \xrightarrow{\mathcal{I}^{\downarrow}, O^{\uparrow}} ms' \text{ with } \mathsf{root}(ms') = l \text{ if and only if } [\![_l\, \mathcal{R}, u\,]\!]_l^i \xrightarrow{\mathcal{I}^{\downarrow}, O^{\uparrow}} [\![_l\, \mathcal{R}, u'\,]\!]_l^i \tag{16}$$

where $ms' = [\![_l\, \mathcal{R}, u'\,]\!]_l^i$, and that:

$$ms \xrightarrow{\mathcal{I}^{\downarrow}, O^{\uparrow}} ms' \text{ with } \mathsf{root}(ms') \neq l \text{ if and only if } [\![_l\, \mathcal{R}, u\,]\!]_l^i \xrightarrow{\mathcal{I}^{\downarrow}, O^{\uparrow} u''} \mathsf{v} \tag{17}$$

where for each set of rules $\overline{\mathcal{R}}$, multisets of objects $\overline{u}$ and $\overline{i}$ and label $\overline{l}$ it holds that:

1. $\mu([_{\bar{l}}\,\overline{\mathcal{R}}, \overline{u}O^{\uparrow}\,]^{\bar{i}}_{\bar{l}}, ms' \mid ms'')$
   $=$
   $\mu([_{\bar{l}}\,\mathsf{FlatOut}'(\overline{\mathcal{R}}, u'' \cap \mathbf{S}, u''O^{\uparrow}\mathsf{obj}(\mathcal{R}) \setminus \mathcal{R}_l) \cup \mathcal{R} \setminus \mathcal{R}_l, \overline{u}u''O^{\uparrow})\,]^{\bar{i}}_{\bar{l}}, ms'')$

2. $\mu([_{\bar{l}}\,\overline{\mathcal{R}}, \overline{u}O^{\uparrow}\,]^{\bar{i}}_{\bar{l}}, ms')$
   $=$
   $[\![_{\bar{l}}\,\mathsf{FlatOut}'(\overline{\mathcal{R}}, u'' \cap \mathbf{S}, u''O^{\uparrow}\mathsf{obj}(\mathcal{R}) \setminus \mathcal{R}_l) \cup \mathcal{R} \setminus \mathcal{R}_l, \overline{u}u''O^{\uparrow})\,]\!]^{\bar{i}}_{\bar{l}}$

Before giving the proof, let us explain (17). If $\mathsf{root}(ms') \neq l$, membrane l dissolves since some rule r in l produces object δ. Analogously, also a rule in $\mathcal{R}$ obtained trough $\mathsf{FlatOut}'$ from r produces δ, thus implying that $[\![_l\,\mathcal{R}, u\,]\!]^i_l$ dissolves. In this case, it sends out the set of objects u'' that correspond, after suitable renaming, to objects in the membranes in ms', plus objects in $\mathbf{S}$ plus objets in $\mathbf{D}$. Consider a system obtained by inserting ms' into any membrane. According to (17), flattening such a system is the same of applying $\mathsf{FlatOut}'$ to rules in the outmost membrane and taking rules in $\mathcal{R} \setminus \mathcal{R}_l$. In fact, this set of rules has been already obtained by repeatedly applying the flattening axiom.

Proof:
We reason by induction over the structure of ms. The case with ms flat is trivial. Hence, let us assume $ms \equiv \mu([_l\,\mathcal{R}', u'\,]^i_l, \bigcup_{i\in I} ms_i)$, with $\mathsf{root}(ms_i) = l_i$ for each $i \in I$.

We prove (16), the proof of (17) is analogous. We prove the "only if" case, the converse is analogous.

The transition $\mu\left([_l\,\mathcal{R}', u'\,]^i_l, \bigcup_{i\in I} ms_i\right) \xrightarrow{\mathcal{I}^{\downarrow}, O^{\uparrow}} ms'$ must be inferred from a transition of each of the $|I|+1$ components of $\mu\left([_l\,\mathcal{R}', u'\,]^i_l, \bigcup_{i\in I} ms_i\right)$. The transition from $[_l\,\mathcal{R}', u'\,]^i_l$ has the following shape, where $I^{\uparrow}$ and $O^{\downarrow}$ are suitable sets of objects, and M is a suitable set of pairs of membrane labels and objects:

$$[_l\,\mathcal{R}', u'\,]^i_l \xrightarrow{M, \mathcal{I}^{\downarrow}, I^{\uparrow}, O^{\uparrow}, O^{\downarrow}} [_l\,\mathcal{R}', v'\,]^i_l \qquad (18)$$

Then, there is a partition $J \cup H$ of I such that for each $i \in J$ the transition from ms_i has either shape

$$ms_i \xrightarrow{\{(l_i, I_i)\}, O^{\uparrow}_i} ms'_i \text{, with } \mathsf{root}(ms'_i) = l_i \qquad (19)$$

or shape

$$ms_i \xrightarrow{\varnothing, O^{\uparrow}_i} ms'_i \text{, with } \mathsf{root}(ms'_i) = l_i \qquad (20)$$

and for each $i \in H$ the transition from ms_i has either shape

$$ms_i \xrightarrow{\{(l_i, I_i)\}, O^{\uparrow}_i} ms'_i \text{, with } \mathsf{root}(ms'_i) \neq l_i \qquad (21)$$

or shape

$$ms_i \xrightarrow{\varnothing, O^{\uparrow}_i} ms'_i \text{, with } \mathsf{root}(ms'_i) \neq l_i \qquad (22)$$

In other words H is the set of the labels of the membranes ms_i performing a step in which a dissolving rule is applied, and J is the set of the labels of the membranes ms_i which do not dissolve.

Since cases (20) and (22) are easier to manage than cases (19) and (21), we consider only these latter two cases. We assume that both $J \neq \emptyset$ and $H \neq \emptyset$. The cases in which either J or H is empty are simpler and can be handled in a slightly different way. It holds that (19) and (21) originate

$$\bigcup_{i \in I} ms_i \xrightarrow{\bigcup_{i\in I}\{(l_i,I_i)\},\bigcup_{i\in I} O_i^{\uparrow}} \bigcup_{i\in I} ms_i' \tag{23}$$

through semantic rules $(jux1) - (jux3)$. The transition from $\mu\left([_l\, \mathcal{R}', u'\,]_l^i\,, \bigcup_{i\in I} ms_i\right)$ is inferred through semantic rule $(h1)$ from (18) and (23), and takes the shape

$$\mu([_l\, \mathcal{R}', u'\,]_l^i\,, \bigcup_{i\in I} ms_i) \xrightarrow{\mathcal{I}^{\downarrow}, O^{\uparrow}} \mu([_l\, \mathcal{R}', v'\,]_l^i\,, \bigcup_{i\in I} ms_i') \tag{24}$$

and it holds that:

1. $O^{\downarrow} = \bigcup_{i\in I}\{(l_i, I_i)\}$;
2. $I^{\uparrow} = \bigcup_{i\in I} O_i^{\uparrow}$;
3. $\nexists (l,a) \in M.\, l \in \bigcup_{i\in I}\{l_i\} \wedge a \in \mathsf{Interface}(l_i, ms_i)$.

Note that in point number 1 we have $=$ rather than $\doteq$ because we have assumed that transitions performed by each ms_i have (19) and (21) as possible shapes.

For each $i \in I$, let us take the set of rules $\mathcal{R}_i$, the set of objects u_i and the set of objects j_i such that $ms_i = [\![_{l_i}\, \mathcal{R}_i, u_i\,]\!]_{l_i}^{j_i}$ is inferred from the flattening axiom.
For each $i \in J$, by the inductive hypothesis it follows that (19) implies

$$[\![_{l_i}\, \mathcal{R}_i, u_i\,]\!]_{l_i}^{j_i} \xrightarrow{\{(l_i,I_i)\},O_i^{\uparrow}} [\![_{l_i}\, \mathcal{R}_i, u_i'\,]\!]_{l_i}^{j_i} \tag{25}$$

where $ms_i' = [\![_{l_i}\, \mathcal{R}_i, u_i'\,]\!]_{l_i}^{j_i}$.
For each $i \in H$, by the inductive hypothesis it follows that (21) implies

$$[\![_{l_i}\, \mathcal{R}_i, u_i\,]\!]_{l_i}^{j_i} \xrightarrow{\{(l_i,I_i)\},O_i^{\uparrow} u_i''} \mathsf{v} \tag{26}$$

where for each $\overline{\mathcal{R}}$, $\overline{u}$, $\overline{l}$ and $\overline{i}$:
$\mu([_{\overline{l}}\, \overline{\mathcal{R}}, \overline{u} O_i^{\uparrow}\,]_{\overline{l}}^{\overline{i}}\,, ms_i' \mid ms'')$
$=$
$\mu([_{\overline{l}}\, \mathsf{FlatOut}'(\overline{\mathcal{R}}, u_i'' \cap \mathbf{S}, u_i'' O_i^{\uparrow} \mathsf{obj}(\mathcal{R}_i) \setminus \mathcal{R}_{l_i}) \cup \mathcal{R}_i \setminus \mathcal{R}_{l_i}, \overline{u} u_i'' O_i^{\uparrow}\,]_{\overline{l}}^{\overline{i}}, ms'')$

Transition (18) is inferred either through $(m1)$, if $\mathcal{I}^{\downarrow} = \varnothing$, or through $(m2)$, if $\mathcal{I}^{\downarrow} = (l, I)$ for some I. The two cases are similar. Let us consider only the case $(m2)$. The originating transition has the shape

$$(\mathcal{R}', u') \xrightarrow[v,U,v,v',D]{M,I\,I^{\uparrow},O^{\uparrow},\bigcup_{i\in I}\{(l_i,I_i)\}} (\mathcal{R}', v') \tag{27}$$

for suitable v, U, v' and D. Notice that $\mathsf{Set}(I) \subseteq i$.
Transitions (25) and (26) are inferred through semantic rules $(fm1)$ and $(fm2)$, respectively, from transitions having the following shape:

$$[_{l_i}\, \mathcal{R}_i, u_i\,]_{l_i}^{j_i} \xrightarrow{\varnothing,\{(l_i,I_i)\},\varnothing,O_i^{\uparrow},\varnothing} [_{l_i}\, \mathcal{R}_i, u_i'\,]_{l_i}^{j_i} \tag{28}$$

$$[_{l_i}\, \mathcal{R}_i, u_i\,]^{j_i}_{l_i} \xrightarrow{\varnothing,\{(l_i,I_i)\},\varnothing,O_i^{\uparrow}u_i'',\varnothing} \mathsf{v} \tag{29}$$

In turn, (28) is inferred through $(m2)$, and (29) is inferred through $(m4)$. The originating transitions have the shape

$$(\mathcal{R}_i, u_i) \xrightarrow[v_i,U_i,v_i,v_i',D_i]{\varnothing,I_i,O_i^{\uparrow},\varnothing} (\mathcal{R}_i, u_i') \tag{30}$$

$$(\mathcal{R}_i, u_i) \xrightarrow[v_i,U_i,v_i,v_i',D_i]{\varnothing,I_i,O_i^{\uparrow}u_i'',\varnothing} \mathsf{v} \tag{31}$$

for suitable values v_i, U_i, v_i', D_i.
Let $\mathcal{R}'' = \mathsf{FlatOut}'(\mathcal{R}', \bigcup_{i\in I}\{\mathbf{s}_{\sigma\cdot l_i} \mid \mathbf{s}_\sigma \in u_i\} \bigcup_{i\in I}\mathbf{s}_{l_i}, \bigcup_{i\in I}\mathsf{Rid}(\mathsf{obj}(\mathcal{R}_i)u_i, l_i))$. Note that for each transition $v \to v_h v_o\{v_{l_i}\}|_D$ in $\mathcal{R}'$, function occ generates several transitions of the form $v' \to v_h v_h' v_o|_{DD'}$, with v' and v_h' derived from v and $\{v_{l_i}\}$, respectively, and D' depending on the v' chosen and on $\{v_{l_i}\}$. However, only transitions generated from transitions of the form $v \to v_h v_o\{v_{l_i}\}|_D$ such that each l_i is in $\bigcup_{i\in I}\{l_i\}$ can fire, since transitions generated from transitions sending objects to membranes contained in any l_i require promoter $\mathbf{d}_{l_i}$, which is not available. Moreover, for the same reason, among the transitions generated from $v \to v_h v_o\{v_{l_i}\}|_D$, only the transition $v \to v_h\{\mathsf{Rid}(v_{l_i}, l_i)\}v_o|_{DD'}$ with v as left side can fire. In this case, it holds that $D' \subseteq \{\neg\mathbf{d}_{l_i} | i \in I\} \cup \bigcup_{i\in I}\{\hat{a}_{l_i} | a \in V\}$. From (27) and the definition of occ we infer

$$(\mathcal{R}'', u' \bigcup_{i\in I}\{\mathbf{s}_{l_i}\} \bigcup_{i\in I}\mathsf{Hat}(j_i, l_i)) \xrightarrow[v,U,v,v',DD']{\varnothing,I\,I^{\uparrow},O^{\uparrow},\varnothing} (\mathcal{R}'', v' \bigcup_{i\in I}\{\mathbf{s}_{l_i}\} \bigcup_{i\in I}\mathsf{Hat}(j_i, l_i) \bigcup_{i\in I}\mathsf{Rid}(I_i, l_i)) \tag{32}$$

where $D' \subseteq \{\neg\mathbf{d}_{l_i} | i \in I\}$. Notice that D' does not contain any $\hat{a}_{l_i}$ with $a \in I_i$. In fact, all these objects are contained in $\mathsf{Hat}(j_i, l_i)$. By removing all $O_i^{\uparrow}$ from $I^{\uparrow}$ and v', from (32) we infer

$$(\mathcal{R}'', u' \bigcup_{i\in I}\{\mathbf{s}_{l_i}\} \bigcup_{i\in I}\mathsf{Hat}(j_i, l_i)) \xrightarrow[v,U,v,v',DD']{\varnothing,I,O^{\uparrow},\varnothing} (\mathcal{R}'', (v' \bigcup_{i\in I}\{\mathbf{s}_{l_i}\} \bigcup_{i\in I}\mathsf{Hat}(j_i, l_i) \bigcup_{i\in I}\mathsf{Rid}(I_i, l_i)) \setminus \bigcup_{i\in I} O_i^{\uparrow}) \tag{33}$$

For each $i \in I$, let $\mathcal{R}_i' = \mathsf{FlatIn}'(\mathcal{R}_i, l_i)$. For each $i \in J$, from (30) and by removing input I_i we infer:

$$(\mathcal{R}_i', \mathsf{Rid}(u_i, l_i)) \xrightarrow[\hat{v}_i,\overline{U_i},\overline{v_i},\overline{v_i'},\mathsf{Rid}(D_i,l_i)\neg\mathbf{d}_{l_i}]{\varnothing,\varnothing,\varnothing,\varnothing} (\mathcal{R}_i', O_i^{\uparrow}\mathsf{Rid}(u_i' \setminus I_i, l_i)) \tag{34}$$

where $\overline{v_i}, \overline{U_i}, \overline{v_i'}$ denote $\mathsf{Rid}(v_i, l_i)$, $\mathsf{Rid}(U_i, l_i)$, $\mathsf{Rid}(v_i', l_i)$, respectively.
For each $i \in H$, from (31) and by removing input I_i we infer:

$$(\mathcal{R}_i', \mathsf{Rid}(u_i, l_i)) \xrightarrow[\overline{v_i},\overline{U_i},\overline{v_i},\overline{v_i'},\mathsf{Rid}(D_i,l_i)\neg\mathbf{d}_{l_i}]{\varnothing,\varnothing,\varnothing,\varnothing} (\mathcal{R}_i', u_i''O_i^{\uparrow} \setminus \mathsf{Rid}(I_i, l_i)) \tag{35}$$

where $\overline{v_i}, \overline{U_i}, \overline{v_i'}$ denote $\mathsf{Rid}(v_i, l_i)$, $\mathsf{Rid}(U_i, l_i)$, $\mathsf{Rid}(v_i', l_i)$, respectively. Notice that u_i'' contains $\mathbf{d}_{l_i}$.

Through semantic rule $(u1)$, from (33), (34) and (35) we infer

$$(\mathcal{R}'' \cup \bigcup_{i\in I} \mathcal{R}'_i, u' \bigcup_{i\in I}\{\mathbf{s}_{l_i}\} \bigcup_{i\in I} \mathsf{Hat}(j_i,l_i) \bigcup_{i\in I} \mathsf{Rid}(u_i,l_i)) \xrightarrow[v\overline{v_1},U\oplus\overline{U_1},v\overline{v_1},v'\overline{v'_1},D\bigcup_{i\in I}\mathsf{Rid}(D_i,l_i)\{\neg\mathbf{d}_{l_i}|i\in I\}]{\varnothing,I,O^{\uparrow},\varnothing} (\mathcal{R}'' \cup \bigcup_{i\in I} \mathcal{R}'_i, v' \bigcup_{i\in I}\{\mathbf{s}_{l_i}\} \bigcup_{i\in I} \mathsf{Hat}(j_i,l_i) \bigcup_{i\in J} \mathsf{Rid}(u'_i,l_i) \bigcup_{i\in H} u''_i) \quad (36)$$

where $\overline{U_1} = \bigcup_{i\in I} \overline{U_i}$, $\overline{v_1} = \bigcup_{i\in I} \overline{v_i}$, $\overline{v'_1} = \bigcup_{i\in I} \overline{v'_i}$.
Let us observe now that $\mathcal{R}'' \cup \bigcup_{i\in I} \mathcal{R}'_i = \mathcal{R}$ and $u' \bigcup_{i\in I} \mathbf{s}_{l_i} \bigcup_{i\in I} \mathsf{Hat}(j_i,l_i) \bigcup_{i\in I} \mathsf{Rid}(u_i,l_i) = u$. Hence, (36) can be rewritten as

$$(\mathcal{R}, u) \xrightarrow[v\overline{v_1},U\oplus\overline{U_1},v\overline{v_1},v'\overline{v'_1},D\bigcup_{i\in I}\mathsf{Rid}(D_i,l_i)\{\neg\mathbf{d}_{l_i}|i\in I\}]{\varnothing,I,O^{\uparrow},\varnothing} (\mathcal{R}, v' \bigcup_{i\in I} \mathsf{Hat}(j_i,l_i) \bigcup_{i\in I} \mathbf{s}_{l_i} \bigcup_{i\in J} \mathsf{Rid}(u'_i,l_i) \bigcup_{i\in H} u''_i) \quad (37)$$

By applying semantic rule $(m2)$, which is applicable since we already know that $\mathsf{Set}(I) \subseteq i$, we infer:

$$[_l\, \mathcal{R}, u\,]^i_l \xrightarrow{\varnothing,I,\varnothing,O^{\uparrow},\varnothing} [_l\, \mathcal{R}, v' \bigcup_{i\in I} \mathsf{Hat}(j_i,l_i) \bigcup_{i\in I} \mathbf{s}_{l_i} \bigcup_{i\in J} \mathsf{Rid}(u'_i,l_i) \bigcup_{i\in H} u''_i\,]^i_l \quad (38)$$

Finally, by applying semantic rule ($fm1$) we obtain

$$[\![_l\, \mathcal{R}, u\,]\!]^i_l \xrightarrow{\mathcal{I}^{\downarrow},O^{\uparrow}} [\![_l\, \mathcal{R}, v' \bigcup_{i\in I} \mathsf{Hat}(j_i,l_i) \bigcup_{i\in I} \mathbf{s}_{l_i} \bigcup_{i\in J} \mathsf{Rid}(u'_i,l_i) \bigcup_{i\in H} u''_i\,]\!]^i_l \quad (39)$$

Take now the target terms of (24) and (39). By the inductive hypothesis over all $i \in J$, the target term of (24) is equated to

$$\mu([_l\, \mathcal{R}', v'\,]^i_l\,, \bigcup_{i\in J} [\![_{l_i}\, \mathcal{R}_i, u'_i\,]\!]^{j_i}_{l_i} \cup \bigcup_{i\in H} ms'_i)$$

By the hypothesis over all $i \in H$, we infer that this is equal to

$$\mu([_l\, R^*, v' \bigcup_{i\in H} u''_i\,]^i_l, \bigcup_{i\in J} [\![_{l_i}\, \mathcal{R}_i, u'_i\,]\!]^{j_i}_{l_i})$$

where $R^* = \mathsf{FlatOut}'(\mathcal{R}', \bigcup_{i\in H} u''_i \cap \mathbf{S}, \bigcup_{i\in H} u''_i O^{\uparrow}_i \mathsf{obj}(\mathcal{R}_i)) \cup \bigcup_{i\in H} \mathcal{R}_i$. This is equated to

$$[\![_l\, R^*_1 R^*_2, v' \bigcup_{i\in H} u''_i \bigcup_{i\in J} \mathbf{s}_{l_i} \bigcup_{i\in J} \mathsf{Hat}(j_i,l_i) \bigcup_{i\in J} \mathsf{Rid}(u'_i,l_i)\,]\!]^i_l$$

where

1. $R^*_1 = \mathsf{FlatOut}'(\mathcal{R}^*, \bigcup_{i\in J}\{\mathbf{s}_{\sigma\cdot l_i} \mid \mathbf{s}_\sigma \in u'_i\} \bigcup_{i\in J} \mathbf{s}_{l_i}, \bigcup_{i\in J} \mathsf{Rid}(u'_i \mathsf{obj}(\mathcal{R}_i), l_i))$;
2. $R^*_2 = \bigcup_{i\in J} \mathsf{FlatIn}'(\mathcal{R}_i, l_i)$.

In the target term of (39) we can remove all rules in $\mathcal{R}$ originally in membranes $\{l_i|i \in H\}$ by exploiting axioms $(ax1)$ and $(ax2)$. This is possible since u_i'' contains $\{\mathbf{d}_{l_i}|i \in H\}$. Then, we can remove also the objects $\bigcup_{i\in H} \mathbf{s}_{l_i}$ and $\bigcup_{i\in H} \mathsf{Hat}(j_i, l_i)$ with axiom $(ax9)$, thus obtaining that in the target terms of (24) and (39) we can have the same set of objects, modulo axioms. We aim to show that the same holds for the set of rules. We have that R_1^* is the union of $\mathsf{FlatOut}'(\mathcal{R}', \bigcup_{i\in J}\{\mathbf{s}_{\sigma\cdot l_i} \mid \mathbf{s}_\sigma \in u_i'\} \bigcup_{i\in J} \mathbf{s}_{l_i} \cup \bigcup_{i\in H} u_i'' \cap \mathbf{S}, \bigcup_{i\in J} \mathsf{Rid}(u_i' \mathsf{obj}(\mathcal{R}_i), l_i) \cup \bigcup_{i\in H} u_i'' O_i^{\uparrow} \mathsf{obj}(\mathcal{R}_i) \setminus \mathcal{R}_{l_i})$ and $\mathsf{FlatOut}'(\bigcup_{i\in H} \mathcal{R}_i \setminus \mathcal{R}_{l_i}, \bigcup_{i\in J}\{\mathbf{s}_{\sigma\cdot l_i} \mid \mathbf{s}_\sigma \in u_i'\} \bigcup_{i\in J} \mathbf{s}_{l_i}, \bigcup_{i\in J} \mathsf{Rid}(u_i' \mathsf{obj}(\mathcal{R}_i), l_i))$. The first set is equal to $\mathcal{R}''$, provided that from $\mathcal{R}''$ we remove promoters $\mathbf{d}_{l_i}$, for all $i \in H$, with axioms $(ax3)$ and $(ax4)$. The second set is equal to $\bigcup_{i\in H} \mathcal{R}_i \setminus \mathcal{R}_{l_i}$, namely it is not affected by $\mathsf{FlatOut}'$. So, it suffices to prove that for each $i \in H$ we can transform $\mathsf{FlatIn}'(\mathcal{R}_i \setminus \mathcal{R}_{l_i}, l_i)$ into $\mathcal{R}_i \setminus \mathcal{R}_{l_i}$. This holds, since all transitions in $\mathsf{FlatIn}'(\mathcal{R}_i, l_i)$ marked with inhibitors appearing in u_i'' can be removed with axioms $(ax1)$ and $(ax2)$.

□

6. 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 [3], the P Algebra, and preserving the behaviour. We have formally defined such a technique first in the case of P Systems without dissolving rules and then in the case of P Systems with dissolving rules. This has required extending the syntax and the semantics of the P Algebra with promoters and inhibitors in evolution rules, with a notion of interface and with a notion of flat membrane, defining some axioms and proving that these axioms are sound with respect to bisimulations.

Future work might include studying the cost of our flattening technique in terms of number of objects and evolution rules added to the flat membrane with respect to the number of objects and evolution rules of the original P System.

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.

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] L. Aceto, W. J. Fokkink and C. Verhoef: Structural Operational Semantics. In: J.A. Bergstra, A. Ponse and S.A. Smolka (Eds.), Handbook of Process Algebra, Elsevier, Amsterdam, 2001, 197–292.

[2] O. Andrei, G. Ciobanu and D. Lucanu: A Rewriting Logic Framework for Operational Semantics of Membrane Systems. Theoret. Comp. Sci. 373, 2007, 163–181.

[3] R. Barbuti, A. Maggiolo–Schettini, P. Milazzo and S.Tini: Compositional Semantics and Behavioral Equivalences for P Systems. Theoret. Comput. Sci. 395, 2008, 77–100.

[4] L. Bianco and V. Manca: Encoding–Decoding Transitional Systems for Classes of P Systems. Proc. Workshop on Membrane Computing (WMC 2005), Lecture Notes in Computer Science 3850, Springer, Berlin, 2006, 134–143.

[5] L. Bianco, F. Fontana, G. Franco and V. Manca: P Systems for Biological Dynamics. Proc. Applications of Membrane Computing, Springer, Berlin, 2006, 83–128.

[6] N. Busi: Using Well–structured Transition Systems to Decide Divergence for Catalytic P Systems. Theoret. Comput. Sci. 372, 2007, 125–135.

[7] N. Busi: Causality in Membrane Systems. Proc. Workshop on Membrane Computing (WMC 2007), Lecture Notes in Computer Science 4860, Springer, Berlin, 2007, 160–171.

[8] R. de Simone: High Level Synchronization Devices in Meije-SCCS. Theoret. Comput. Sci. 37, 1985, 245–267.

[9] R. Freund and S. Verlan: A Formal Framework for Static (Tissue) P Systems. Proc. Workshop on Membrane Computing (WMC 2007), Lecture Notes in Computer Science 4860, Springer, Berlin, 2007, 271–284.

[10] D. Harel and A. Pnueli: On the Development of Reactive Systems. In: K.R. Apt (editor), Logic and Models of Concurrent Systems, NATO, ASI-13, Springer, New York, 1985, 477–498.

[11] R. Milner: Communication and Concurrency. Prentice-Hall, London, 1989.

[12] D. Park: Concurrency and Automata on Infinite Sequences. Proc. Proc. Theoretical Computer Science: 5^{th} GI conference, Lecture Notes in Computer Science 104, Springer, Berlin, 1981, 167–183.

[13] G. Păun: Membrane computing. An introduction, Springer, 2002.

[14] I. Petre: A Normal Form for P-Systems. Bulletin of the EATCS 67 1999, 165–172.

[15] G. Plotkin: A Structural Approach to Operational Semantics. J. Log. Algebr. Program. 60–61, 2004, 17–139.