Compositional Semantics and Behavioral Equivalences for P Systems

Roberto Barbuti, Andrea Maggiolo–Schettini, Paolo Milazzo

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

Simone Tini

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

Abstract

The aim of the paper is to give a compositional semantics in the style of the Structural Operational Semantics (SOS) and to study behavioral equivalence notions for P Systems. Firstly, we consider P Systems with maximal parallelism and without priorities. We define a process algebra, called P Algebra, whose terms model membranes, we equip the algebra with a Labeled Transition System (LTS) obtained through SOS transition rules, and we study how some equivalence notions defined over the LTS model apply in our case. Then, we consider P Systems with priorities and extend the introduced framework to deal with them. We prove that our compositional semantics reflects correctly maximal parallelism and priorities.

Key words: P Systems, Structural Operational Semantics, Behavioral equivalence, Congruence.

1 Introduction

P Systems were introduced by Păun in [16,17] as distributed parallel computing devices inspired by the structure and the functioning of a living cell. The

Email addresses: barbuti@di.unipi.it (Roberto Barbuti), maggiolo@di.unipi.it (Andrea Maggiolo–Schettini), milazzo@di.unipi.it (Paolo Milazzo), simone.tini@uninsubria.it (Simone Tini).

cell is considered as a set of compartments enclosed by membranes. A P System consists of a *hierarchy of membranes*, each of them containing a multiset of *objects*, representing molecules, a set of *evolution rules*, representing chemical reactions, and possibly other membranes. For each evolution rule there are two multisets of objects, describing the reactants and the products of the chemical reaction. A rule in a membrane can be applied only to objects in the same membrane. Some objects produced by the rule remain in the same membrane, others are sent *out* of the membrane, others are sent *into* the inner membranes, which are identified by their labels. Evolution rules are applied with *maximal parallelism*, meaning that it cannot happen that some evolution rule is not applied when the objects needed for its triggering are available. Some rules cause *dissolution* of the membrane. In this case, the objects are put in the membrane containing the dissolved one, and the rules disappear. The outmost membrane, called *skin membrane*, can never dissolve. Finally, evolution rules can be endowed with a priority relation: a rule can be applied only if no rule with higher priority is applicable.

The aim of this paper is to study some notions of *behavioral equivalence* over P Systems, formalizing the idea of membranes having the *same behavior* and being *substitutable* to each other. Often, two computation systems are considered to be equivalent when they have the same input/output behavior with respect to the external world. In the P System setting, the external world of a membrane m is formed by the membranes contained by m and by either the membrane containing m, if m is not the skin, or the external environment, otherwise. Hence, it seems to be reasonable to consider as equivalent two membranes that, at each computation step, can receive the same objects from outer and inner membranes, can send the same objects to the outer membrane or to the external world, and can send the same objects to the same inner membranes. Objects obtained as input and objects produced as output are, therefore, what in the field of concurrency theory is usually considered to be the *observable* part of the behavior of the membrane. Of course, a reasonable notion of equivalence should be a *congruence*, meaning that the equivalence should be preserved when two equivalent membranes are put in the same context, namely when they are inserted into the same outer membrane and host the same inner membranes.

Structural operational semantics (SOS) [18,19] has been demonstrated to be a solid framework for defining equivalence notions over process algebras (see [3] for a survey). The idea is to give the operational semantics of the process algebra in terms of a *labeled transition system* (LTS) [12,19], having a state for each process and a labeled transition for each computation step leading from a process to another process. Transition labels describe what happens in the considered step at a suitable level of detail. The LTS is obtained by means of *transition rules* of the form $\frac{premises}{conclusion}$, describing how a computation step of a process (instance of the conclusion) is derived from computation steps of other processes (instances of premises). Finally, an equivalence relation over the LTS states is defined, so that two processes s_1 and s_2 are equated if and only if the portions of LTS rooted in s_1 and s_2 are the same, provided that some details of the branching structure of the LTS are abstracted away. By changing these details, one has different notions of equivalence.

Following the SOS approach, in this paper we firstly introduce a process algebra, called *P algebra*, whose terms (processes) represent either *membrane contents*, namely multisets of objects and sets of evolution rules that can be composed with other membrane contents to form a membrane, or *membranes*, or *juxtaposition of membranes*, namely sets of membranes that can be inserted into the same outer membrane. For readability, the P Algebra does not consider, in a first phase, priority relations among evolution rules. We equip the P Algebra with an LTS, obtained by means of a set of transition rules. To describe the observable part of membrane behavior, LTS transition labels contain information on input objects and output objects that are received and sent in the computation step. We shall argue that the price we have to pay to build the LTS in an inductive way, i.e. through the transition rules, is that the LTS labels must contain other details on the computation steps that are not related to the input/output behavior. In particular, part of this information is needed to model in a compositional way the maximal parallelism on the application of the evolution rules, which is a global property.

Then, we consider five different notions of behavioral preorder that have been defined in the literature over the LTS model, and the equivalences obtained by considering their kernels. Each of these preorders has a proper abstraction of the LTS branching structure, and gives a proper abstraction on the way in which processes compute. These preorders are structured in a hierarchy of inclusion, meaning that, given two of these preorders, one is (not necessarily strictly) stronger than the other. In general, one can provide LTSs for which these inclusions are strict. What is not obvious at all is whether these inclusions are strict in the LTS inferred from the P Algebra. Actually, we give examples showing that all these inclusions are strict also in our LTS. In some sense, this demonstrates the expressivity of the P System model. We shall prove that all the equivalences obtained from the preorders enjoy the congruence property with respect to all the operations of the P Algebra. This demonstrates the solidity of our proposal, and, moreover, it can supply the basis for the development of axiomatic frameworks in which the equivalences can fit.

Finally, we consider another process algebra, called *Priority P Algebra (PP Algebra)*, which extends the P Algebra to deal with priority. Also in this case the inclusions over the preorders are strict and the equivalences enjoy the congruence property.

Related work. A paper dealing with SOS for P Systems is [5] (whose preliminary version is [4]). In [5] three auxiliary transition relations between process algebra terms are considered, describing application of evolution rules, object communication between membranes, and membrane dissolution, respectively. Transitions describing the whole computation step of the membrane are obtained by applying, in a given order, one transition for each relation. Our approach is different, since we have one only transition relation, representing the whole computation step. This allows us to have a stronger notion of compositionality, since in [5] compositionality is achieved for each of the three auxiliary relations, but not for the main relation. In [5] the LTS describing the evolution steps of the membranes have no label, whereas our labels carry information over inputs and outputs. This permits us to have a notion of observable, and, therefore, notions of equivalence taking into account the observable behavior of membranes. By removing labels from our LTS we could easily obtain the LTS of [5].

Operational semantics for P Systems have been proposed in [8,9,11]. In [8] Catalytic P Systems without priorities are considered and their semantics is given as a Well–Structured Transition System. A step of a system is not constructed compositionally, but by means of an additional transition relation describing the application of single evolution rules. The main purpose of the paper is to prove decidability of the divergence problem for the considered variant of P Systems. The semantics of [8] is extended in [9] to describe the causal dependencies occurring between reactions of a P System. In [11] a formal framework is proposed to describe a large number of variants of P Systems. The framework consists of an operational semantics parameterized by the type of parallelism and halting conditions, and so on. All the semantics in [8,9,11] are not compositional and have no notion of observable behavior.

It is worth to note that the literature offers other formalisms in which the input/output behavior of programs can be defined compositionally only by taking into account additional information, namely by using as transition labels not only the information on the observable part of the behavior. For instance, [13,14,20–22] argue that the input/output behavior of *synchronous languages* [7] can be described compositionally only by exploiting information on the causality relations among single inputs and single outputs.

2 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.

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

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

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 message δ such that $\delta \notin V$:

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

Application of evolution rules is done in parallel, and it could be regulated by *priority* relations between rules. Parallelism is maximal; 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 priority relations are such that a rule with a priority smaller than another cannot be chosen for application if the one with greater priority is applicable. The lower–priority rule cannot be chosen even if the high–priority one is not chosen for application: what really matters is the fact that the latter is applicable. 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

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

been produced.

Now, we formally define P Systems.

Definition 1 *A* P System Π *is given by*

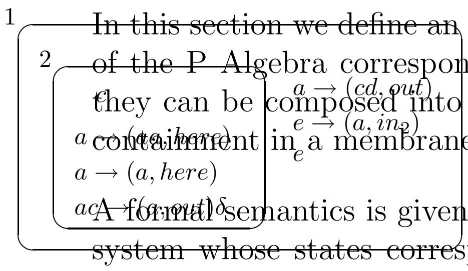
$$\Pi = (V, \mu, w_1, \ldots, w_n, (R_1, \rho_1), \ldots, (R_n, \rho_n))$$

where:

- V *is an* alphabet *whose elements are called* objects*;*
- $\mu \subset \mathbb{N} \times \mathbb{N}$ *is a* membrane structure*, such that* $(i, j) \in \mu$ *denotes that the membrane labeled by* j *is contained in the membrane labeled by* i*;*
- w_i *with* $1 \leq i \leq n$ *are strings from* V^* *representing multisets over* V *associated with the membranes* $1, 2, \ldots, n$ *of* μ*;*
- R_i *with* $1 \leq i \leq n$ *are finite sets of* evolution rules *associated with the membranes* $1, 2, \ldots, n$ *of* μ*;*
- ρ_i *is a partial order relation over* R_i*, specifying a* priority *relation between rules:* $(r_1, r_2) \in \rho_1$ *iff* $r_1 > r_2$ *(i.e.* r_1 *has a higher priority than* r_2*).*

We show in Fig. 1 an example of P System in which all the main features of the formalism are used.

3 The P Algebra



In this section we define an algebra of P Systems, called *P Algebra*. 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.

A formal semantics is given to the P Algebra in terms of a labeled transition system whose states correspond to P Systems and whose transitions correspond to P System computation steps in which evolution rules are applied with maximal parallelism. We remark that our semantics is compositional, namely the semantics of a system is inferred from the semantics of its components. Compositionality allows component–wise reasoning and verification of properties, but it is not easily achievable, in general, in formalisms with global behavioral constraints such as the maximal parallelism of P Systems.

3.1 Syntax

We assume the usual string notation to represent multisets. For instance, to represent $\{a, a, b, b, c\}$ we may write either $aabbc$, or a^2b^2c, or $(ab)^2c$. We denote multiset (and set) union as string concatenation, hence we write u_1u_2 for $u_1 \cup u_2$.

For the sake of readability, we shall write $u \to v_hv_o\{v_{l_i}\}$ for the generic non–dissolving evolution rule $u \to (v_h, here)(v_o, out)(v_1, in_{l_1}) \ldots (v_n, in_{l_n})$, and $u \to v_hv_o\{v_{l_i}\}\delta$ for the similar generic dissolving evolution rule. Moreover, in examples we shall often write $u \to v_hv_ov_{l_1} \ldots v_{l_n}$, where $l_1, \ldots, l_n$ are the child membranes with $l_i < l_{i+1}$ and $v_{l_i} = \varnothing$ if no multiset is sent by the evolution rule into child membrane l_i. Similarly for dissolving rules. For instance, we will write $a \to \varnothing bc\varnothing d$ to denote the rule $a \to (\varnothing, here)(bc, out)(\varnothing, in_1)(d, in_2)$ if the child membranes of the membrane containing the rule are labeled by 1 and 2.

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

Definition 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* $\mathbb{N}$ *and a over V:*

$$\begin{aligned} c &::= (\varnothing, \varnothing) \;\Big|\; (u \to v_hv_o\{v_{l_i}\}, \varnothing) \;\Big|\; (u \to v_hv_o\{v_{l_i}\}\delta, \varnothing) \;\Big|\; (\varnothing, a) \;\Big|\; c \cup c \\ m &::= [_l c]_l \\ ms &::= m \;\Big|\; ms \mid ms \;\Big|\; \mu(m, ms) \;\Big|\; \mathsf{v} \end{aligned}$$

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 trough the union operation $\_\cup\_$ from constants representing single evolution rules and constants representing single objects, and can be plugged into a membrane l by means of the operation $[_l \_]_l$. Formally:

- $(\varnothing, \varnothing)$ represents the empty membrane content;
- $(u \to v_hv_o\{v_{l_i}\}, \varnothing)$, for all $u, v_h, v_o, v_{l_i} \in V^*$, represents the membrane content with a single non-dissolving evolution rule and no object;
- $(u \to v_hv_o\{v_{l_i}\}\delta, \varnothing)$, for all $u, v_h, v_o, v_{l_i} \in V^*$, represents the membrane content with a single dissolving evolution rule and no object;

- $(\varnothing, a)$, for all $a \in V$, represents the membrane content with no evolution rule and a single object;
- $c_1 \cup c_2$: given c_1 representing $(\mathcal{R}_1, u_1)$ and c_2 representing $(\mathcal{R}_2, u_2)$, $c_1 \cup c_2$ represents the union of these membrane contents, namely $(\mathcal{R}_1 \cup \mathcal{R}_2, u_1 u_2)$.

Membrane systems have the following meaning:

- $[_l c\,]_l$: given a membrane content c representing the pair $(\mathcal{R}, u)$ and $l \in \mathbb{N}$, $[_l c\,]_l$ represents the membrane having l as label, $\mathcal{R}$ as evolution rules and u as objects;
- $ms_1 \mid ms_2$: 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;
- v represents the *dissolved membrane.*

Example 3 *Let us consider Fig. 1. The membranes labeled 1 and 2 are represented by the following terms t_1 and t_2, respectively:*

$$t_1 \equiv [_1\ (a \rightarrow \varnothing cd\varnothing, \varnothing) \cup (e \rightarrow \varnothing\varnothing a, \varnothing) \cup (\varnothing, e)\ ]_1$$

$$t_2 \equiv [_2\ (a \rightarrow aa\varnothing, \varnothing) \cup (a \rightarrow a\varnothing, \varnothing) \cup (ac \rightarrow \varnothing a\delta, \varnothing) \cup (\varnothing, c)\ ]_2$$

Moreover, $\mu(t_1, t_2)$ represents the containment of membrane 2 in membrane 1.

Notice that Def. 2 includes membrane systems that have no corresponding P System. In particular, we need v to model the state reached after application of a dissolving evolution rule, and we need juxtaposition to use only one binary hierarchical composition operation instead of an $(n+1)$-ary operation μ_n, for all $n \geq 1$, to represent the containment of n membranes in another one.

Notice also that the definition of μ excludes that the first argument is a juxtaposition $ms_1 \mid ms_2$ of two membrane systems, or a hierarchical composition $\mu(ms_1, ms_2)$ of two membrane systems. Hence, it cannot happen that two juxtaposed membranes have a common child, and that a parent and its child have a common child.

3.2 Maximally Parallel Semantics

Let us recall the model of labeled transition systems [12,19].

Definition 4 (LTS) *A* labeled transition system *(LTS for short) is 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 $(s, s') \in \xrightarrow{\ell}$.

The semantics of the P Algebra is given in terms of an LTS, with a state for each syntactically correct term. LTS labels can be of the following forms:

- $(u, U, v, v', M, I, O^{\uparrow}, O^{\downarrow})$, describing a computation step performed by a membrane content c, where:
  - u is the multiset of objects consumed by the application of evolution rules in c, as it results from the composition, by means of $\_ \cup \_$, of the constants representing these evolution rules.
  - U is the set of multisets of objects corresponding to the left hand sides of the evolution rules in c.
  - v is the multiset of objects in c offered for the application of the evolution rules, as it results from the composition, by means of $\_ \cup \_$, of the constants representing these objects. When operation $[_l \; \_ \;]_l$ is applied to c, it is required that v and u coincide.
  - v' is the multiset of objects in c that are not used to apply any evolution rule and, therefore, are not consumed, as it results from the composition, by means of $\_ \cup \_$, of the constants representing these objects. When operation $[_l \_]_l$ is applied to c, it is required that no multiset in U is contained in v', thus implying that no evolution rule in c can be furtherly applied by exploiting the available objects. This constraint is mandatory to ensure maximal parallelism.
  - M contains a membrane label l if some evolution rule in c is not applied since its firing would imply sending objects to some child membrane labeled l, but no child membrane labeled l exists. When the operation μ is applied to $([_{l'} c]_{l'}, ms)$, for any membrane system ms and membrane label l', it is required that l is not a membrane in ms.
  - I is the multiset of objects received as inputs from the parent membrane and from the child membranes.
  - $O^{\uparrow}$ is the multiset of objects sent as an output to the parent membrane.
  - $O^{\downarrow}$ is a set of pairs (l_i, v_{l_i}) describing the multiset of objects sent as an output to each child membrane l_i.
- $(M, \mathcal{I}, O^{\uparrow}, O^{\downarrow})$, describing a computation step performed by a membrane system ms, where: $\mathcal{I}$ is a set of pairs (l_i, v_{l_i}) describing the multiset of objects received as an input by each membrane l_i in ms, and M, $O^{\uparrow}$ and $O^{\downarrow}$ are as in the previous case.

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

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

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

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

$$\frac{I \in V^* \quad M \subseteq \mathsf{Labels}(\{v_{l_i}\}) \quad M \neq \varnothing}{(u \to v_h v_o\{v_{l_i}\}, \varnothing) \xrightarrow[\varnothing, \varnothing, \varnothing, \varnothing]{M, I, \varnothing, \varnothing} (u \to v_h v_o\{v_{l_i}\}, I)} \quad (mc4)$$

$$\frac{I \in V^* \quad M \subseteq \mathsf{Labels}(\{v_{l_i}\}) \quad M \neq \varnothing}{(u \to v_h v_o\{v_{l_i}\}\delta, \varnothing) \xrightarrow[\varnothing, \varnothing, \varnothing, \varnothing]{M, I, \varnothing, \varnothing} (u \to v_h v_o\{v_{l_i}\}\delta, I)} \quad (mc5)$$

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

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

Fig. 2. Rules for membrane contents.

compositionally only by taking into account more information than input and output themselves [13,14,20–22].

For the sake of legibility, in transitions with labels of the first form we shall write the first four 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 [19] of the form $\frac{premises}{conclusion}$, where the premises are a set of transitions, and the conclusion is a transition. Intuitively, SOS transition rules permits us to infer moves of P Algebra terms from moves of their subterms. We assume the standard way to associate a set of transitions with a set of transition rules [3].

In Fig. 2 we introduce the rules for membrane contents. For each $n \geq 0$, rule $(mc1_n)$ describes n simultaneous applications of the evolution rule $u \to v_h v_o\{v_{l_i}\}$. In the label, u^n means that n occurrences of the multiset of objects u are consumed, $\{u\}$ means that there is an evolution rule able to consume u, I are the objects received from child membranes and the parent membrane, v_o^n means that n occurrences of v_o are sent to the parent membrane, and $(l_i, v_{l_i}^n)$ means that n occurrences of v_{l_i} are sent to each child membrane l_i. In the target state, I and v_h^n mean that the objects received as inputs and the n occurrences of v_h produced by the n applications of the evolution rule will be available at the next computation step.

The dissolving evolution rule $u \to v_h v_o\{v_{l_i}\}\delta$ requires two types of semantic rules. For each $n \geq 1$, rule $(mc2_n)$ describes n simultaneous applications of the dissolving rule, leading to dissolution. Note that in this case all I, v_o^n, and v_h^n are sent as an output to the parent membrane, and that the target state is v. Moreover, also δ appears among the outputs, so that also the label carries information about dissolution. The second semantic rule is $(mc3)$, which describes the case in which the dissolving rule is not applied. In this case δ does not appear in the label and the objects in I will be available at the next computation step.

In rules $(mc4)$ and $(mc5)$ we assume a function Labels from sets of pairs $\{(v_1, in_{l_1}), \ldots, (v_n, in_{l_n})\}$, 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 $\{l_1, \ldots, l_n\}$. Rule $(mc4)$ states that the evolution rule $u \to v_h v_o\{v_{l_i}\}$ is not applied because it requires sending objects to child membranes in M that are assumed not to exist. Rule $(mc5)$ is analogous to $(mc4)$, but deals with dissolving evolution rules.

Rules $(mc6)$ and $(mc7)$ describe the behavior of the objects inside the membrane content. Rule $(mc6)$ states that object a is offered for the application of some evolution rule. Rule $(mc7)$ states that object a is not used to apply any evolution rule. In this second case, a appears in the target state and will be available at the next computation step. Finally, $(mc8)$ expresses that the empty membrane can only receive input.

In Fig. 3 we introduce the transition rules allowing to infer behavior of unions of membrane contents from the behavior of the individual membrane contents. 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, if 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 $\{u \in U_1U_2 \mid \nexists u' \in U_1U_2.u' \subset u\}$. Finally, given two sets $O_1^{\downarrow}$ and $O_2^{\downarrow}$ representing two outputs to inner membranes, we write $O_1^{\downarrow} \cup_{\mathbb{IN}} O_2^{\downarrow}$ to denote the set $\{(l, uv) \mid (l, u) \in O_1^{\downarrow} \wedge (l, v) \in O_2^{\downarrow}\} \cup \{(l, u) \mid (l, u) \in O_1^{\downarrow} \wedge \nexists v.(l, v) \in O_2^{\downarrow}\} \cup \{(l, v) \mid (l, v) \in O_2^{\downarrow} \wedge \nexists u.(l, u) \in O_1^{\downarrow}\}$.

Rule $(u1)$ describes the move of $x_1 \cup x_2$ when no dissolving rule is applied ($\delta \notin O_1^{\uparrow}O_2^{\uparrow}$). Rule $(u2)$ describes the case in which x_1 is dissolved due to a dissolving rule, and x_2 is not ($\delta \in O_1^{\uparrow}$ and $\delta \notin O_2^{\uparrow}$). Since x_1 and x_2 will be plugged into the same membrane, also x_2 has to dissolve and all its objects, denoted $\mathsf{Objects}(y_2)$, are sent to the parent membrane. We assume a rule analogous to

$$\frac{x_1 \xrightarrow[u_1, U_1, v_1, v_1']{M_1, I_1, O_1^{\uparrow}, O_1^{\downarrow}} y_1 \quad x_2 \xrightarrow[u_2, U_2, v_2, v_2']{M_2, I_2, O_2^{\uparrow}, O_2^{\downarrow}} y_2 \quad \begin{array}{c} M_1M_2 \cap \mathsf{Labels}(O_1^{\downarrow} \cup_{\mathbb{N}} O_2^{\downarrow}) = \varnothing \\ v_1'v_2' \not\vdash U_1 \oplus U_2 \quad \delta \notin O_1^{\uparrow}O_2^{\uparrow} \end{array}}{x_1 \cup x_2 \xrightarrow[u_1u_2, U_1 \oplus U_2, v_1v_2, v_1'v_2']{M_1M_2, I_1I_2, O_1^{\uparrow}O_2^{\uparrow}, O_1^{\downarrow} \cup_{\mathbb{N}} O_2^{\downarrow}} y_1 \cup y_2} \qquad (u1)$$

$$\frac{x_1 \xrightarrow[u_1, U_1, v_1, v_1']{M_1, I_1, O_1^{\uparrow}, O_1^{\downarrow}} y_1 \quad x_2 \xrightarrow[u_2, U_2, v_2, v_2']{M_2, I_2, O_2^{\uparrow}, O_2^{\downarrow}} y_2 \quad \begin{array}{c} M_1M_2 \cap \mathsf{Labels}(O_1^{\downarrow} \cup_{\mathbb{N}} O_2^{\downarrow}) = \varnothing \\ v_1'v_2' \not\vdash U_1 \oplus U_2 \quad \delta \in O_1^{\uparrow} \quad \delta \notin O_2^{\uparrow} \end{array}}{x_1 \cup x_2 \xrightarrow[u_1u_2, U_1 \oplus U_2, v_1v_2, v_1'v_2']{M_1M_2, I_1I_2, O_1^{\uparrow}O_2^{\uparrow}\mathsf{Objects}(y_2), O_1^{\downarrow} \cup_{\mathbb{N}} O_2^{\downarrow}} \mathsf{v}} \qquad (u2)$$

$$\frac{x_1 \xrightarrow[u_1, U_1, v_1, v_1']{M_1, I_1, O_1^{\uparrow}, O_1^{\downarrow}} y_1 \quad x_2 \xrightarrow[u_2, U_2, v_2, v_2']{M_2, I_2, O_2^{\uparrow}, O_2^{\downarrow}} y_2 \quad \begin{array}{c} M_1M_2 \cap \mathsf{Labels}(O_1^{\downarrow} \cup_{\mathbb{N}} O_2^{\downarrow}) = \varnothing \\ v_1'v_2' \not\vdash U_1 \oplus U_2 \quad \delta \in O_1^{\uparrow} \cap O_2^{\uparrow} \end{array}}{x_1 \cup x_2 \xrightarrow[u_1u_2, U_1 \oplus U_2, v_1v_2, v_1'v_2']{M_1M_2, I_1I_2, O_1^{\uparrow}O_2^{\uparrow}, O_1^{\downarrow} \cup_{\mathbb{N}} O_2^{\downarrow}} \mathsf{v}} \qquad (u3)$$

Fig. 3. Rules for union of membrane contents.

$(u2)$ to deal with the symmetric case, namely x_2 is dissolved and x_1 is not. Rule $(u3)$ describes the case in which both x_1 and x_2 are dissolved ($\delta \in O_1^{\uparrow} \cap O_2^{\uparrow}$).

In all the three rules it is required that $M_1M_2 \cap \mathsf{Labels}(O_1^{\downarrow} \cup_{\mathbb{N}} O_2^{\downarrow}) = \varnothing$, namely that the set of child membranes x_1 (resp. x_2) assumes to be absent has a null intersection with the set of child membranes to which objects should be sent by x_2 (resp. x_1). Moreover, it is required that $v_1'v_2' \not\vdash U_1 \oplus U_2$, namely that objects that are not consumed by x_1 and x_2 cannot trigger any evolution rule in x_1 and x_2. Both these requirements are needed to guarantee maximal parallelism of evolution rules. Note that $U_1 \oplus U_2$ does not contain any multiset u if it contains some multiset $u' \subset u$, since objects triggering u would trigger also u'. We anticipate that this allows considering as equivalent systems such as $(a \to \varnothing b \varnothing, \varnothing) \cup (a \to \varnothing c \varnothing, \varnothing) \cup (aa \to \varnothing bc \varnothing, \varnothing)$ and $(a \to \varnothing b \varnothing, \varnothing) \cup (a \to \varnothing c \varnothing, \varnothing)$, which represent the contents of the membranes shown in Fig. 4. If we replace $U_1 \oplus U_2$ with U_1U_2 in $(u1)$, $(u2)$, $(u3)$, we obtain that the two systems perform transitions which differ only in the element of the label corresponding to U_1U_2. In particular, the first system would perform transitions in which such an element is $\{a\}$ while the second system would perform transitions in which such an element is $\{a, aa\}$. This unwanted situation is avoided by the use of $U_1 \oplus U_2$.

Among the transitions performed by membrane contents, we call *acceptable* those corresponding to actual evolution rule applications, namely those in which the multiset of objects consumed by evolution rules and the multiset of objects offered for their application, do coincide.

Fig. 4. An example of equivalent systems.

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

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

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

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

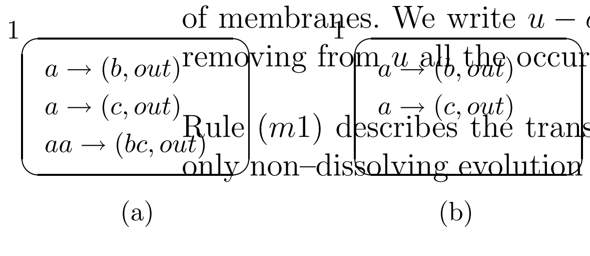
Fig. 5. Rules for single membranes and juxtaposition of membranes.

Lemma 5 *Operation ∪ on membrane contents is associative and commutative.*

PROOF. Operation ∪ on membrane contents is commutative because the two transition rules $(u1)$ and $(u3)$ are completely symmetric, while rule $(u2)$, which is not symmetric, has a dual version, omitted in Figure 3. Moreover, operation ∪ on membrane contents is also associative because the labels of the transitions in the conclusions of $(u1)$, $(u2)$ and $(u3)$ are constructed by using associative operations, namely set and multiset unions, and because the transition rules $(mc1_n), \ldots, (mc5)$ describe both the cases of application and non–application of any evolution rule, and the transition rules $(mc5), \ldots, (mc8)$ describe both the cases in which the objects are consumed and are not consumed. □

In Fig. 5 we give the transition rules for single membranes and for juxtaposition of membranes. We write $u - \delta$ to denote the multiset of objects obtained by removing from u all the occurrences of δ.

Rule $(m1)$ describes the transition in a membrane with content x and when only non–dissolving evolution rules are applied. Among the transitions x may



$$\frac{x_1 \xrightarrow{M_1,\{(l_1,I_1)\},O_1^{\uparrow},O_1^{\downarrow}} y_1 \quad x_2 \xrightarrow{M_2,\mathcal{I}_2,O_2^{\uparrow},\varnothing} y_2 \quad \begin{array}{c} O_1^{\downarrow} \doteqdot \mathcal{I}_2 \quad O_2^{\uparrow} \subseteq I_1 \\ M_1 \cap \mathsf{Labels}(\mathcal{I}_2) = \varnothing \quad \delta \notin O_1^{\uparrow}O_2^{\uparrow} \end{array}}{\mu(x_1,x_2) \xrightarrow{\varnothing,(l_1,I_1 \backslash O_2^{\uparrow}),O_1^{\uparrow},\varnothing} \mu(y_1,y_2)} \quad (h1)$$

$$\frac{x_1 \xrightarrow{M_1,\{(l_1,I_1)\},O_1^{\uparrow},O_1^{\downarrow}} y_1 \quad x_2 \xrightarrow{M_2,\mathcal{I}_2,O_2^{\uparrow},\varnothing} y_2 \quad \begin{array}{c} O_1^{\downarrow} \doteqdot \mathcal{I}_2 \quad O_2^{\uparrow} \subseteq I_1 \quad \delta \in O_1^{\uparrow} \\ M_1 \cap \mathsf{Labels}(\mathcal{I}_2) = \varnothing \ \ \delta \notin O_2^{\uparrow} \end{array}}{\mu(x_1,x_2) \xrightarrow{\varnothing,\{(l_1,I_1 \backslash O_2^{\uparrow})\},O_1^{\uparrow}-\delta,\varnothing} y_2} \quad (h2)$$

$$\frac{x_1 \xrightarrow{M_1,\{(l_1,I_1)\},O_1^{\uparrow},O_1^{\downarrow}} y_1 \quad x_2 \xrightarrow{M_2,\mathcal{I}_2,O_2^{\uparrow},\varnothing} y_2 \quad \begin{array}{c} O_1^{\downarrow} \doteqdot \mathcal{I}_2 \ \ O_2^{\uparrow} - \delta \subseteq I_1 \ \ \delta \notin O_1^{\uparrow} \\ M_1 \cap \mathsf{Labels}(\mathcal{I}_2) = \varnothing \quad \delta \in O_2^{\uparrow} \end{array}}{\mu(x_1,x_2) \xrightarrow{\varnothing,\{(l_1,I_1 \backslash O_2^{\uparrow})\},O_1^{\uparrow},\varnothing} y_1} \quad (h3)$$

$$\frac{x_1 \xrightarrow{M_1,\{(l_1,I_1)\},O_1^{\uparrow},O_1^{\downarrow}} y_1 \quad x_2 \xrightarrow{M_2,\mathcal{I}_2,O_2^{\uparrow},\varnothing} y_2 \quad \begin{array}{c} O_1^{\downarrow} \doteqdot \mathcal{I}_2 \ \ O_2^{\uparrow} - \delta \subseteq I_1 \\ M_1 \cap \mathsf{Labels}(\mathcal{I}_2) = \varnothing \quad \delta \in O_1^{\uparrow} \cap O_2^{\uparrow} \end{array}}{\mu(x_1,x_2) \xrightarrow{\varnothing,\{(l_1,I_1 \backslash O_2^{\uparrow})\},O_1^{\uparrow},\varnothing} \mathsf{v}} \quad (h4)$$

Fig. 6. Rules for hierarchy of membranes.

perform, only the acceptable ones are considered (the first and the third components of the label below the arrow coincide). This condition, together with constraint $v_1'v_2' \nvdash U_1 \oplus U_2$ in rules $(u1)$, $(u2)$ and $(u3)$, allows describing maximal parallelism compositionally. The input I to the content x becomes the input (l, I) to the membrane l. Rule $(m2)$ describes the case in which dissolving evolution rules are applied.

Rule $(jux1)$ describes the juxtaposition of two membranes that do not dissolve. Since Def. 2 ensures that $x_1 \mid x_2$ cannot appear as the first argument of operation μ, namely x_1 and x_2 cannot have any common child, we are sure that the set of children of x_1 and x_2 cannot furtherly change. Hence, we can assume that both x_1 and x_2 have delivered all outputs to their child membranes, namely that the set of remaining outputs to be delivered to inner membranes is empty, as it appears from the fourth component of the label of both premises, which is empty. For the same reason, information in M_1 and M_2 is needless, and, therefore, the first component of the label of the conclusion is set to $\varnothing$. Rules $(jux2)$ and $(jux3)$ describe the juxtaposition when one or both of the two component membranes dissolve. We assume also a rule symmetric to $(jux2)$.

In Fig. 6 we introduce the transition rules concerning hierarchical composition of membranes. 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 $(\mathcal{I}_1 \setminus \{(l, \varnothing) \mid l \in \mathbb{N}\}) = (\mathcal{I}_2 \setminus \{(l, \varnothing) \mid l \in \mathbb{N}\})$.

All these rules describe the case in which the membrane system x_2 is contained in the membrane x_1. The fact that x_1 is a single membrane emerges by looking at the second element of the label of the transitions that x_1 performs (namely $\{(l_1, I_1)\}$), which represents inputs x_1 expects and is a single pair. To handle correctly object exchange between parent membrane and child membranes, we require that the multisets of objects $O_1^{\downarrow}$ sent from parent to children is equivalent to the multisets of objects $\mathcal{I}_2$ expected as input by the children. We also require that the objects $O_2^{\uparrow}$ sent from children to parent are contained in the objects I_1 expected as input by the parent. Moreover, to check that the children expected to be absent by x_1 are really absent, we require that the labels in M_1 do not appear in $\mathcal{I}_2$. Objects in $O_2^{\uparrow}$ are removed from I_1 in the label of the transition of the hierarchical composition, hence the input objects in the composition are only those expected as input from outside the resulting membrane. Rules $(h1)$, $(h2)$, $(h3)$, and $(h4)$ handle the cases in which either no membrane, or only the the parent membrane, or only all the child membranes, or all of them dissolve, respectively.

3.3 Properties of the Semantics

First of all let us note that all SOS rules respect the well known *de Simone* format [10], i.e. they have the form

$$\frac{\{x_i \xrightarrow{\alpha_i} y_i \,|i \in I\}}{f(x_1, \ldots, x_{ar(f)}) \xrightarrow{\alpha} t}$$

where f is an operation with arity $ar(f)$, $x_1, \ldots, x_{ar(f)}$ and $\{y_i \,|\, i \in I\}$ are variables that can be instantiated with any term, I is any subset of $\{1, \ldots, ar(f)\}$, the variables x_i and y_j are all distinct and the only variables that occur in the rule, and, finally, t is any term that does not contain any variable x_i, for $i \in I$, and has no multiple occurrence of variables.

To prove that our SOS style semantics reflects maximal parallelism, we show that if a membrane content $(\mathcal{R}, u)$ performs an arbitrary acceptable transition $(\mathcal{R}, u) \xrightarrow[u',U,u',v']{M,I,O_1^{\uparrow},O_1^{\downarrow}} x$, then the multiset of objects v' that are not consumed in the transition is such that none of the evolution rules in $\mathcal{R}$ can be applied to its objects, i.e. the membrane content $(\mathcal{R}, v')$ is not able to perform any acceptable transition $(\mathcal{R}, v') \xrightarrow[u'',U',u'',v'']{M,I',O_2^{\uparrow},O_2^{\downarrow}}$ with $u'' \neq \varnothing$, for any $U', v'', I', O_2^{\uparrow}, O_2^{\downarrow}$.

Theorem 6 *If* $(\mathcal{R}, u) \xrightarrow[u',U,u',v']{M,I,O_1^{\uparrow},O_1^{\downarrow}} x$*, then* $(\mathcal{R}, v') \not\xrightarrow[u'',U',u'',v'']{M,I',O_2^{\uparrow},O_2^{\downarrow}}$ *for any* $u'' \neq \varnothing$ *and any* $U', v'', I', O_2^{\uparrow}, O_2^{\downarrow}$.

PROOF. We prove the following stronger implication:

$$(\mathcal{R}, u) \xrightarrow[u',U,v,v']{M,I,O_1^{\uparrow},O_1^{\downarrow}} x \implies (\mathcal{R}, v') \not\xrightarrow[u'',U',u'',v'']{M,I',O_2^{\uparrow},O_2^{\downarrow}} \text{ for any } u'' \neq \varnothing \qquad (1)$$

in which the assumption is a transition non necessarily acceptable (i.e. with v non necessarily equal to u').

The proof of (1) is by induction on the cardinality of $\mathcal{R}$.

- Base cases:
  - If $\mathcal{R} = \varnothing$ then all transitions from $(\mathcal{R}, v')$ are inferred from rules $(mc6)$, $(mc7)$, $(mc8)$ and $(u1)$. Hence, those that are acceptable have the form $(\mathcal{R}, v') \xrightarrow[\varnothing,\varnothing,\varnothing,v']{\varnothing,I,\varnothing,\varnothing}$, and (1) is trivially satisfied.
  - If $\mathcal{R}$ contains a single non–dissolving rule, namely $\mathcal{R} = \{u_1 \to v_h v_o\{v_{l_i}\}\}$, we can assume, by Lemma 5, that

    $$(u_1 \to v_h v_o\{v_{l_i}\}, u) = (u_1 \to v_h v_o\{v_{l_i}\}, \varnothing) \cup (\varnothing, u)\,.$$

    The premise of (1) can be rewritten as

    $$(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing) \cup (\varnothing, u) \xrightarrow[u',U,v,v']{M,I,O_1^{\uparrow},O_1^{\downarrow}} x\,.$$

    The only way to derive this transition is by applying rule $(u1)$ with $(\varnothing, u)$ as x_1 and $(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing)$ as x_2 (or vice–versa). All the transitions that can be performed by $(\varnothing, u)$ are inferred through rules $(mc6)$, $(mc7)$ and $(u1)$, and have the form

    $$(\varnothing, u) \xrightarrow[\varnothing,\varnothing,\overline{u}_1,\overline{u}_2]{\varnothing,I_2,\varnothing,\varnothing} (\varnothing, I_2\overline{u}_2)$$

    for $I_2 \in V^*$ and $\overline{u}_1, \overline{u}_2$ such that $\overline{u}_1\overline{u}_2 = u$. The transition performed by $(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing)$ is obtained by applying either rule $(mc1_n)$ or rule $(mc4)$. Let us distinguish these two cases.

    In the case of $(mc1_n)$, the premise of (1) is obtained by applying rule

$(u1)$ to the transitions

$$(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing) \xrightarrow[u_1^n, \{u_1\}, \varnothing, \varnothing]{\varnothing, I_1, O_1^{\uparrow}, O_1^{\downarrow}} (u_1 \to v_h v_o\{v_{l_i}\}, \varnothing),\ (\varnothing, u) \xrightarrow[\varnothing, \varnothing, v, v']{\varnothing, I_2, \varnothing, \varnothing} (\varnothing, I_2 v')$$

Hence, in the premise of (1) we have that M is $\varnothing$, I is $I_1 I_2$, $O_1^{\uparrow}$ and $O_2^{\uparrow}$ depend only on the transition of $(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing)$, u' is u_1^n, for some $n > 0$, U is $\{u_1\}$, and v and v' are such that $vv' = u$. By the premises of rule $(u1)$ we are sure that $v' \nvdash \{u_1\}$, which implies that $u_1 \not\subseteq v'$. Now, by contradiction, let us assume that there is a y such that we can infer a transition $(u_1 \to v_h v_o\{v_{l_i}\}, v') \xrightarrow[u'', U', u'', v'']{M, I', O_2^{\uparrow}, O_2^{\downarrow}} y$ with $u'' \neq \varnothing$. This transition should be inferred from rule $(u1)$. Moreover, the move of $(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing)$ used in the premise of $(u1)$ is inferred from $(mc1_m)$, for some $m > 0$, since, if it was inferred from $(mc4)$ we would have that $u'' = \varnothing$. This implies that u'' is equal to u_1^m, for some $m > 0$. Finally, the move of $(\varnothing, v')$ used in the premise of $(u1)$ determines the value of v'', which is such that $u''v'' = v'$. Summarizing, we have that $u'' = u_1^m$, for some $m > 0$, $u''v'' = v'$ and $u_1 \not\subseteq v'$, which is a contradiction. Hence, the conclusion of implication (1) follows.

In the case of $(mc4)$, the premise of (1) is obtained by applying rule $(u1)$ to the transitions

$$(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing) \xrightarrow[\varnothing, \varnothing, \varnothing, \varnothing]{M, I_1, \varnothing, \varnothing} (u_1 \to v_h v_o\{v_{l_i}\}, \varnothing),\ (\varnothing, u) \xrightarrow[\varnothing, \varnothing, v, v']{\varnothing, I_2, \varnothing, \varnothing} (\varnothing, I_2 v')$$

Hence, in the premise of (1) we have that M is determined by the move of $(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing)$, I is $I_1 I_2$, $O_1^{\uparrow}$, $O_1^{\downarrow}$, u' and U are $\varnothing$, and v and v' are such that $vv' = u$. By the premises of rule $(mc4)$ we have that M is a non empty subset of the labels appearing in $\{v_{l_i}\}$. The same M is assumed in the label of the transition appearing in the conclusion of (1). Now, by contradiction let us assume that there is a y such that we can infer a transition $(u_1 \to v_h v_o\{v_{l_i}\}, v') \xrightarrow[u'', U', u'', v'']{M, I', O_2^{\uparrow}, O_2^{\downarrow}} y$ with $u'' \neq \varnothing$. This transition should be inferred from rule $(u1)$. Moreover, the move of $(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing)$ used in the premise of $(u1)$ is inferred from $(mc4)$, since, if it was inferred from $(mc1_n)$, we would have that $M = \varnothing$. This implies that $u'' = \varnothing$, which is a contradiction. Hence, the conclusion of (1) follows.

· The case in which $\mathcal{R}$ contains a single dissolving rule is similar to the previous case.

- Induction cases:

· If $\mathcal{R} = \{u_1 \to v_h v_o\{v_{l_i}\}\} \cup \mathcal{R}'$, with $\mathcal{R}' \neq \varnothing$, we can assume, by Lemma 5,

that
$$(\mathcal{R}, u) = (\{u_1 \to v_h v_o \{v_{l_i}\}\}, \varnothing) \cup (\mathcal{R}', u)\,.$$
The premise of (1) can be rewritten as
$$(\{u_1 \to v_h v_o \{v_{l_i}\}\}, \varnothing) \cup (\mathcal{R}', u) \xrightarrow[u',U,v,v']{M,I,O_1^\uparrow,O_1^\downarrow} x\,.$$
This transition can be derived by applying either rule $(u1)$ or rule $(u2)$. We consider only the case of $(u1)$, as the case of $(u2)$ is analogous. Rule $(u1)$ can be applied with $(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing)$ as x_1 and $(\mathcal{R}', u)$ as x_2 (or vice–versa), where the transition performed by $(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing)$ and used in the premise of $(u1)$ is obtained by applying either rule $(mc1_n)$ or rule $(mc4)$. Let us distinguish these two cases.

In the case of $(mc1_n)$, the premise of (1) is obtained by applying rule $(u1)$ to the transitions
$$(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing) \xrightarrow[u_1^n,\{u_1\},\varnothing,\varnothing]{\varnothing,I_1,O_a^\uparrow,O_a^\downarrow} (u_1 \to v_h v_o\{v_{l_i}\}, \varnothing),\ (\mathcal{R}', u) \xrightarrow[\overline{u},\overline{U},v,v']{M,I_2,O_b^\uparrow,O_b^\downarrow} x''$$
for some x'' such that $x = (u_1 \to v_h v_o\{v_{l_i}\}, \varnothing) \cup x''$. Hence, in the premise of (1) we have that M, v and v' are determined by the move of $(\mathcal{R}', u)$, $I = I_1 I_2$, $O_1^\uparrow = O_a^\uparrow O_b^\uparrow$, $O_1^\downarrow = O_a^\downarrow \cup_{\mathbb{N}} O_b^\downarrow$, $u' = u_1^n$, for some $n \geq 0$, and $U = \{u_1\} \oplus \overline{U}$. By the premises of rule $(u1)$ we have that $v' \nvdash \overline{U} \oplus \{u_1\}$, which implies that $u_1 \not\subseteq v'$. By the induction hypothesis we have that the transition on the right above implies that
$$(\mathcal{R}', v') \not\xrightarrow[\widetilde{u},\widetilde{U},\widetilde{u},\widetilde{v}']{M,\widetilde{I},\widetilde{O^\uparrow},\widetilde{O^\downarrow}} \qquad \text{for any } \widetilde{u} \neq \varnothing$$
and this implies that with $\widetilde{u} \neq \varnothing$ we have only transitions from $(\mathcal{R}', v')$ of the form
$$(\mathcal{R}', v') \xrightarrow[\widetilde{u},\widetilde{U},\widetilde{v},\widetilde{v}']{M,\widetilde{I},\widetilde{O^\uparrow},\widetilde{O^\downarrow}} y \tag{2}$$
with $\widetilde{u} \neq \widetilde{v}$. These transitions are obtained by applying either rule $(u1)$ or rule $(u2)$, and, by Lemma 5, we can assume that the premises of these rules are a move by $(\mathcal{R}', \varnothing)$ and a move by $(\varnothing, v')$. Moreover, the values of $\widetilde{v}$ and $\widetilde{v}'$ are determined by the move of $(\varnothing, v')$, and there is a move for each each pair $\widetilde{v}$, $\widetilde{v}'$ such that $\widetilde{v}\widetilde{v}' = v'$. We are sure that $\widetilde{u} \not\subseteq v'$. In fact, if $\widetilde{u} \subseteq v'$, we could choose as premise of $(u1)$ the move by $(\varnothing, v')$ with $\widetilde{v} = \widetilde{u}$, which contradicts that $\widetilde{v} \neq \widetilde{u}$. Summarizing, we have proved up to now that $u_1 \not\subseteq v'$ and, if there is a transition like that in (2), either $\widetilde{u} = \varnothing$, or $\widetilde{u} \not\subseteq v'$. We note that all the transitions performed by $(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing) \cup (\mathcal{R}', v')$, that is $(\mathcal{R}, v')$, are inferred by rule $(u1)$ or $(u2)$, where the premises of these rules are both a move by $(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing)$ and a move by $(\mathcal{R}', v')$. The first of these two moves is inferred by either

$(mc1_n)$ or $(mc4)$. Hence we have two subcases.
In the first subcase, the transition by $(\mathcal{R}, v')$ has the form

$$(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing) \cup (\mathcal{R}', v') \xrightarrow[\widetilde{u}u_1^m,\widetilde{U},\widetilde{v},\widetilde{v'}]{M,\widetilde{I},\widetilde{O^{\uparrow}},\widetilde{O^{\downarrow}}} x$$

for some $m \geq 0$, and is inferred from

$$(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing) \xrightarrow[u_1^m,\{u_1\},\varnothing,\varnothing]{\varnothing,I_a,O_a^{\uparrow},O_a^{\downarrow}} (u_1 \to v_h v_o\{v_{l_i}\}, \varnothing),\ (\mathcal{R}', v') \xrightarrow[\widetilde{u},\widetilde{U'},\widetilde{v},\widetilde{v'}]{M,I_b,O_b^{\uparrow},O_b^{\downarrow}} z$$

where $\widetilde{I} = I_a \cup I_b$, $\widetilde{O^{\uparrow}} = O_a^{\uparrow} \cup O_b^{\uparrow}$, $\widetilde{O^{\downarrow}} = O_a^{\downarrow} \cup_{\mathbb{N}} O_b^{\downarrow}$ and $\widetilde{U} = \{u_1\} \oplus \widetilde{U}'$. We have already proved that either $\widetilde{u} = \varnothing$ or $\widetilde{u} \not\subseteq v'$, and that $u_1 \not\subseteq v'$. Morevoer, we know that $\widetilde{v} \subseteq v'$. As a consequence, $\widetilde{u}u_1^m \neq \widetilde{v}$.
In the second subcase, the transition by $(\mathcal{R}, v')$ has the form

$$(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing) \cup (\mathcal{R}', v') \xrightarrow[\widetilde{u},\widetilde{U},\widetilde{v},\widetilde{v'}]{M,\widetilde{I},\widetilde{O^{\uparrow}},\widetilde{O^{\downarrow}}} x$$

and is inferred from

$$(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing) \xrightarrow[\varnothing,\varnothing,\varnothing,\varnothing]{M',I_a,\varnothing,\varnothing} (u_1 \to v_h v_o\{v_{l_i}\}, \varnothing),\ (\mathcal{R}', v') \xrightarrow[\widetilde{u},\widetilde{U},\widetilde{v},\widetilde{v'}]{M'',I_b,\widetilde{O^{\uparrow}},\widetilde{O^{\downarrow}}} z$$

where $\widetilde{I} = I_a \cup I_b$, and $M = M' \cup M''$. Now, if we have the transition $(\mathcal{R}', v') \xrightarrow[\widetilde{u},\widetilde{U},\widetilde{v},\widetilde{v'}]{M'',I_b,\widetilde{O^{\uparrow}},\widetilde{O^{\downarrow}}} z$ then we have also the transition $(\mathcal{R}', v') \xrightarrow[\widetilde{u},\widetilde{U},\widetilde{v},\widetilde{v'}]{M,I_b,\widetilde{O^{\uparrow}},\widetilde{O^{\downarrow}}} z$. This follows from the fact that transition $(\mathcal{R}', u) \xrightarrow[\overline{u},\overline{U},v,v']{M,I_2,O_b^{\uparrow},O_b^{\downarrow}} x''$ assumed at the beginning of the proof, ensures that all labels in M are mentioned by the rules in $\mathcal{R}'$, and rules $(mc4)$ and $(mc5)$ permit to choose an arbitrary subset of the labels mentioned in the evolution rules. If we have the transition $(\mathcal{R}', v') \xrightarrow[\widetilde{u},\widetilde{U},\widetilde{v},\widetilde{v'}]{M,I_b,\widetilde{O^{\uparrow}},\widetilde{O^{\downarrow}}} z$, we have already proved that $\widetilde{u} \not\subseteq v'$. Since we know also that $\widetilde{v} \subseteq v'$, we infer that $\widetilde{u} \neq \widetilde{v}$.

In the case of $(mc4)$ the premise of (1) is obtained by applying rule $(u1)$ to the transitions

$$(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing) \xrightarrow[\varnothing,\varnothing,\varnothing,\varnothing]{M_1,I_1,\varnothing,\varnothing} (u_1 \to v_h v_o\{v_{l_i}\}, \varnothing),\ (\mathcal{R}', u) \xrightarrow[u',U,v,v']{M_2,I_2,O_1^{\uparrow},O_1^{\downarrow}} x''$$

for some x'' such that $x = (u_1 \to v_h v_o\{v_{l_i}\}, \varnothing) \cup x''$. Hence, in the premise of (1) we have that $M = M_1 M_2$, $I = I_1 I_2$, and $O_1^{\uparrow}$, $O_1^{\downarrow}$, u', U, v and v' are determined by the move by $(\mathcal{R}', u)$. Without loss of generality, we can assume that no evolution rule in $\mathcal{R}'$ mentions membrane labels in $M_1 \setminus M_2$. In fact, if some membrane label l is in $M_1 \setminus M_2$ and is mentioned by some evolution rule in $\mathcal{R}'$, then we can replace M_2 with $M_2 \cup \{l\}$ in the transition in the right side. This property holds since rules $(mc4)$ and $(mc5)$ permit to choose an arbitrary subset of the labels mentioned in the evolution rules. By the definition of rule $(mc4)$ we have that $M_1 \cap \mathsf{Labels}(\{v_{l_i}\}) \neq \varnothing$. Since $M = M_1M_2$, it follows that $M \cap \mathsf{Labels}(\{v_{l_i}\}) \neq \varnothing$. By contradiction, let us assume that there is a y such that we can infer a transition $(\mathcal{R}, v') \xrightarrow[u'',U,u'',v'']{M,I',O_2^{\uparrow},O_2^{\downarrow}} y$ with $u'' \neq \varnothing$. Since $M \cap \mathsf{Labels}(\{v_{l_i}\}) \neq \varnothing$, we are sure that this transition can be inferred only by composing through (u_1) or (u_2) a transition by $(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing)$ inferred from rule $(mc4)$ of the form $(u_1 \to v_h v_o\{v_{l_i}\}, \varnothing) \xrightarrow[\varnothing,\varnothing,\varnothing,\varnothing]{M',I',\varnothing,\varnothing,} $ with $M' \subseteq M \cap \mathsf{Labels}(\{v_{l_i}\})$, and a transition by $(\mathcal{R}', v')$ of the form $(\mathcal{R}', v') \xrightarrow[u'',U,u'',v'']{M_2',I',O_2^{\uparrow},O_2^{\downarrow}} y'$, where M_2' is such that $M = M'M_2'$. Since we have assumed that the evolution rules in $\mathcal{R}'$ mention membrane labels in $M_1 \setminus M_2$, if we have the transition $(\mathcal{R}', v') \xrightarrow[u'',U,u'',v'']{M_2',I',O_2^{\uparrow},O_2^{\downarrow}} y'$ we have also the transition $(\mathcal{R}', v') \xrightarrow[u'',U,u'',v'']{M_2,I',O_2^{\uparrow},O_2^{\downarrow}} y'$. But, by the induction hypothesis, we have that $(\mathcal{R}', v') \not\xrightarrow[u'',U,u'',v'']{M_2,I',O_2^{\uparrow},O_2^{\downarrow}}$ for any $u'' \neq \varnothing$. Hence, the conclusion of (1) follows.

· The case in which $\mathcal{R} = \{u_1 \to v_h v_o\{v_{l_i}\}\delta\} \cup \mathcal{R}'$ is similar to the previous case. $\square$

4 Behavioral Preorders and Equivalences

In this section we consider some well known notions of behavioral *preorder* and *equivalence* defined in the literature over the LTS model (see [3] for a survey). Let us recall that a preorder is a reflexive and transitive relation, and an equivalence is a symmetric preorder. Moreover, the largest equivalence contained in a preorder is called the *kernel* of the preorder.

As usual, given an LTS, we shall write $s \stackrel{\ell}{\not\to}$ if $s \xrightarrow{\ell} s'$ holds for no s', and $s \not\to$ if $s \stackrel{\ell}{\not\to}$ for all $\ell \in \mathcal{L}$. Moreover, for a state $s \in \mathcal{S}$, we denote by $\mathsf{Initials}(s)$ the set $\{\ell \in \mathcal{L} \mid \exists s'.\ s \xrightarrow{\ell} s'\}$.

Definition 7 *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* simulation *if, for each pair $s_1 \, R \, s_2$, if $s_1 \xrightarrow{\ell} s_1'$ then there is a transition $s_2 \xrightarrow{\ell} s_2'$ such that $s_1' \, R \, s_2'$;*
- *is a* ready simulation *if it is a simulation and, for each pair $s_1 \, R \, s_2$, if $s_1 \stackrel{\ell}{\not\to}$ then $s_2 \stackrel{\ell}{\not\to}$;*
- *is a* ready trace preorder *if, for each pair $s_1 R s_2$, any ready trace of s_1 is a ready trace of s_2 (a sequence $L_0\ell_1L_1 \ldots \ell_nL_n$ with $L_i \subseteq \mathcal{L}$ and $\ell_i \in \mathcal{L}$ is a*

ready trace *of a state* s_0 *if* $s_0 \xrightarrow{\ell_1} s_1 \xrightarrow{\ell_2} \dots s_{n-1} \xrightarrow{\ell_n} s_n$ *and* $\mathsf{Initials}(s_i) = L_i$ *for* $i = 0, \dots, n$*);*

- *is a* failure preorder *if, for each pair* $s_1 R s_2$*, any failure of* s_1 *is a failure of* s_2 *(a pair* $(\ell_1 \dots \ell_n, L)$ *with* $\ell_1 \dots \ell_n \in \mathcal{L}$ *and* $L \subseteq \mathcal{L}$ *is a* failure *of a state* s *if* $s \xrightarrow{\ell_1} \dots \xrightarrow{\ell_n} s'$ *for some state* s' *such that* $\mathsf{Initials}(s') \cap L = \varnothing$*);*
- *is a* trace preorder *if, for each pair* $s_1 R s_2$*, any trace of* s_1 *is a trace of* s_2 *(a sequence* $\ell_1 \dots \ell_n$ *with* $\ell_i \in \mathcal{L}$ *is a* trace *of a state* s_0 *if* $s_0 \xrightarrow{\ell_1} \dots \xrightarrow{\ell_n} s_n$ *for some state* s_n*).*

All the relations in Def. 7 are preorders. Intuitively, two states s and t are related by some preorder (resp. by an equivalence obtained as kernel of some preorder) if the behavior of s is simulated by (resp. equivalent to) the behavior of t, provided that some details of the behaviors of s and t are abstracted away. These details depend on the considered preorder (resp. equivalence).

As usual, we denote with $\sqsubseteq_{RS}$ (resp.: $\sqsubseteq_S$) the union of all ready simulations (resp.: simulations), which, in turn, is a ready simulation (resp. simulation). The kernel of $\sqsubseteq_{RS}$ and the kernel of $\sqsubseteq_S$ coincide, is called *bisimulation*, and is denoted by $\approx$. We denote by $\sqsubseteq_{RT}$ (resp.: $\sqsubseteq_F$, $\sqsubseteq_T$) the union of all ready trace preorders (resp.: failure preorders, trace preorders), which, in turn, is a ready trace preorder (resp.: failure preorder, trace preorder), and by $\approx_{RT}$ (resp.: $\approx_F$, $\approx_T$) its kernel.

Example 8 *Let us consider the membranes in Fig. 4, which are modeled by the following P Algebra terms:*
$s \equiv [_1\, (a \to \varnothing b, \varnothing) \cup (a \to \varnothing c, \varnothing) \cup (aa \to \varnothing bc, \varnothing)\,]_1$
$t \equiv [_1\, (a \to \varnothing b, \varnothing) \cup (a \to \varnothing c, \varnothing)\,]_1$.
LTS transitions take into account that membrane 1 can be inserted into a parent membrane, from which membrane 1 can receive objects a, b, and c. For each $p, q, r \in \mathbb{N}$ *there exist an LTS state* $s_{p,q,r}$*, and an LTS state* $t_{p,q,r}$*, representing membrane 1 in the left side and membrane 1 in the right side, containing the multiset of objects* $a^p b^q c^r$*. Then, for each* p', q', r'*, and for each pair of values* p_b *and* p_c *such that* $p_b + p_c = p$*, there are transitions* $s_{p,q,r} \xrightarrow{\varnothing,(1,a^{p'}b^{q'}c^{r'}),b^{p_b}c^{p_c},\varnothing} s_{p',q+q',r+r'}$ *and* $t_{p,q,r} \xrightarrow{\varnothing,(1,a^{p'}b^{q'}c^{r'}),b^{p_b}c^{p_c},\varnothing} t_{p',q+q',r+r'}$ *modeling that the membrane exploits the* p *objects* a *to send out* p_b *objects* b *and* p_c *objects* c*, and that the input from the parent membrane is* $a^{p'}b^{q'}c^{r'}$*. Finally, we have transitions* $s \xrightarrow{\varnothing,(1,a^{p'}b^{q'}c^{r'}),b^0c^0,\varnothing} s_{p',q',r'}$ *and* $t \xrightarrow{\varnothing,(1,a^{p'}b^{q'}c^{r'}),b^0c^0,\varnothing} t_{p',q',r'}$*. It turns out that* s *and* t *are equated by the stronger equivalence we have considered, namely* $\approx$*, which reflects the intuition that the membranes in Fig. 4 have the same behavior.*

The previous example shows a case in which one of the evolution rules in a membrane is useless. In Fig. 7 we show other pairs of membranes whose corresponding P Algebra terms are equated by $\approx$. Let us consider the membranes in Fig. 7.a and Fig. 7.b. They have different structure, but have the same observable behavior. Both membranes labeled 1 can receive occurrences of d and a, provided that they are inserted in a parent membrane. Each occurrence of d is send out at the next step. Each occurrence of a is sent inside membrane 2. Now, both membranes labeled 2 are able to sent out one occurrence of d for each occurrence of a they receive from 1, with a delay of three computation steps. Such occurrences of d are sent out by 1 after one more computation step.

Let us consider the membranes in Fig. 7.c and Fig. 7.d. Both membranes labeled 1 can receive occurrences of d and a, provided that they are inserted in a parent membrane. Each occurrence of d is sent out at the next step. For each occurrence of a, the membrane 1 in the left side sends either one occurrence of b inside 2, or an occurrence of c inside 3, whereas the membrane 2 in the right side sends an occurrence of e inside 2. At the next step, the membrane 1 in the left side receives one d for each b sent to 2 and one d for each c sent to 3, whereas the membrane 2 in the right side receives one d for each e sent to 2. These d's are sent out by 1 one step later.

Let us consider the membranes in Fig. 7.e and Fig. 7.f. Both membranes labeled 1 can receive occurrences of b and a, provided that they are inserted in a parent membrane. Each occurrence of b is sent out at the next step. For each occurrence of a, the membrane 1 in the left side sends two occurrences of a inside 2, whereas the membrane 1 in the right side sends two occurrences of a inside 3. Hence, 2 and 3 always receive $2k$ occurrences of a, for some k, and send out to 1 $2k$ occurrences of b at the next step. These b's are sent out by 1 at the next step.

4.1 Congruence Property

A classical requirement over a preorder (resp. equivalence) notion is to be a precongruence (resp. congruence), namely that it is preserved by all operations. This is essential to allow substitution of programs with equivalent ones and to develop an axiomatic framework.

Definition 9 *A preorder (resp. equivalence) $R \subseteq \mathcal{S} \times \mathcal{S}$ is called a* precongruence *(resp.* congruence*) iff, for each operation f with arity n and pairs $s_1 R s'_1, \dots s_n R s'_n$, it holds that $f(s_1, \dots, s_n) R f(s'_1, \dots, s'_n)$.*

All P Algebra operations behave well w.r.t. preorders and equivalences in Def. 7.

Theorem 10 *All preorders in Def. 7 are precongruences.*

Fig. 7. Examples of bisimilar systems.

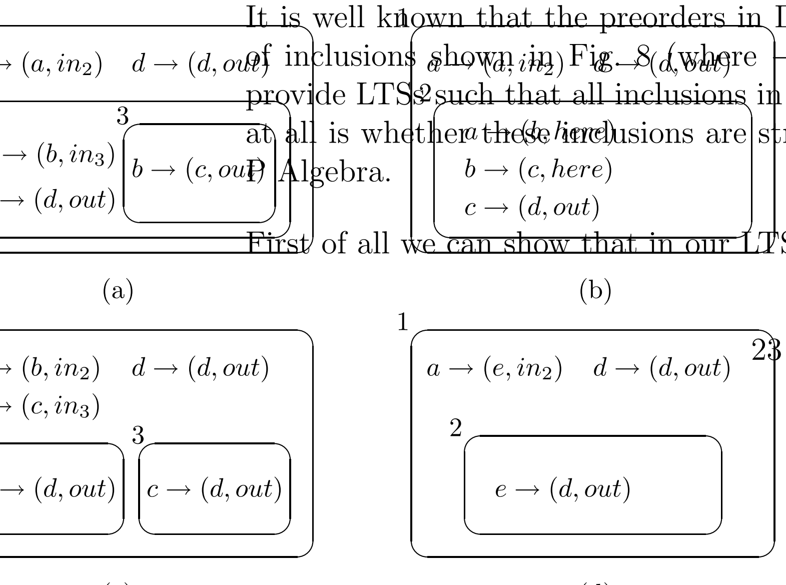
PROOF. In [23] it is proved that trace preorder and failure preorder are precongruences for all calculi defined with SOS rules in de Simone format. In [24] some formats are given which ensure that simulation preorder, ready simulation preorder and ready trace preorder are precongruences. All these formats are less restrictive than de Simone format. □

Corollary 11 *The kernels of all preorders in Def. 7 are congruences.*

4.2 Hierarchy of Preorders

It is well known that the preorders in Def. 7 are structured by the hierarchy of inclusions shown in Fig. 8 (where $\mapsto$ stands for $\subseteq$). In general, one can provide LTSs such that all inclusions in Fig. 8 are strict. What is not obvious at all is whether these inclusions are strict also in the LTS inferred from the P Algebra.

First of all we can show that in our LTS the inclusion of $\sqsubseteq_{RS}$ in $\sqsubseteq_S$ is strict.



$$\begin{tikzcd}
& \sqsubseteq_S \arrow[rrd] & & \\
\sqsubseteq_{RS} \arrow[ru] \arrow[rd] & & & \sqsubseteq_T \\
& \sqsubseteq_{RT} \arrow[rru] \arrow[r] & \sqsubseteq_F &
\end{tikzcd}$$

Fig. 8. Relations between preorders

Example 12 *Let us consider the following P Algebra terms:*
$s \equiv [_1\,(\varnothing, a) \cup (a \to \varnothing b\delta, \varnothing)\,]_1$
$t \equiv [_1\,(\varnothing, a) \cup (a \to \varnothing b\delta, \varnothing) \cup (a \to \varnothing c\delta, \varnothing)\,]_1$.
Term s represents a membrane that can send out b and dissolve, whereas term t represents a membrane that can send out either b, or c, and dissolve. Formally, for each $p, q, r \in \mathbb{N}$, $s \xrightarrow{l_b^{p,q,r}} \mathsf{v}$, and $t \xrightarrow{l_b^{p,q,r}} \mathsf{v}$ and $t \xrightarrow{l_c^{p,q,r}} \mathsf{v}$ are the transitions in the parts of the LTS that are rooted in s and t, respectively, where $l_b = (\varnothing, (1, a^p b^q c^r), a^p b^{q+1} c^r, \varnothing)$ and $l_c = (\varnothing, (1, a^p b^q c^r), a^p b^q c^{r+1}, \varnothing)$ represent the computation steps performed by the membrane for input $a^p b^q c^r$. Note that the input is sent out due to dissolution. Now, we have that $s \sqsubseteq_S t$, since t is able to simulate each $l_b^{p,q,r}$ move by s, but $s \not\sqsubseteq_{RS} t$, since $t \xrightarrow{l_c^{p,q,r}} \mathsf{v}$ and s has no $l_c^{p,q,r}$ move.

Now we can show that both inclusions of $\sqsubseteq_S$ in $\sqsubseteq_T$ and $\sqsubseteq_{RS}$ into $\sqsubseteq_{RT}$ are strict.

Example 13 *Let us consider the membranes in Fig. 9. Their behavior differ since the membrane in the left side chooses at the first computation step if one object b or one object c will be sent out at the fourth step, whereas the membrane in the right side makes the decision in the second computation step. Actually, in the left side the choice is made by membrane 2, which can send a to either 3 or 4. In the former case 3 sends out b to 2, which sends out b to 1, which sends out b. In the latter case 4 sends out c to 2, which sends out c to 1, which sends out c. In the right side, 2 can only send a to 5, which decides to send out to 2 either b or c. In the former case 2 sends out b to 1, which sends out b. In the latter case 2 sends out c to 1, which sends out c.*

Let us take the following P Algebra terms:
$s_3 \equiv [_3\,(a \to \varnothing b, \varnothing)\,]_3$
$s_4 \equiv [_4\,(a \to \varnothing c, \varnothing)\,]_4$
$s_2 \equiv [_2\,(\varnothing, a) \cup (a \to \varnothing\varnothing a\varnothing, \varnothing) \cup (a \to \varnothing\varnothing\varnothing a, \varnothing) \cup (b \to \varnothing b\varnothing\varnothing, \varnothing) \cup (c \to \varnothing c\varnothing\varnothing, \varnothing)\,]_2$
$t_5 \equiv [_5\,(a \to \varnothing b, \varnothing) \cup (a \to \varnothing c, \varnothing)\,]_5$
$t_2 \equiv [_2\,(\varnothing, a) \cup (a \to \varnothing\varnothing a, \varnothing) \cup (b \to \varnothing b\varnothing, \varnothing) \cup (c \to \varnothing c\varnothing, \varnothing)\,]_2$
$s_1 \equiv t_1 \equiv [_1\,(b \to \varnothing b\varnothing, \varnothing) \cup (c \to \varnothing c\varnothing, \varnothing)\,]_1$
$s \equiv \mu(s_1, \mu(s_2, s_3 \mid s_4))$

Fig. 9. Two membrane systems showing that both inclusions $\sqsubseteq_S \subseteq \sqsubseteq_T$ and $\sqsubseteq_{RS} \subseteq \sqsubseteq_{RT}$ are strict. The labels are as follows: $l^{p',q'}_{1,p,q} = (\varnothing, (1, b^{p'}c^{q'}), b^p c^q, \varnothing)$, $l^{p',q'}_{2,p,q} = (\varnothing, (1, b^{p'}c^{q'}), b^{p+1}c^q, \varnothing)$, $l^{p',q'}_{3,p,q} = (\varnothing, (1, b^{p'}c^{q'}), b^p c^{q+1}, \varnothing)$.

$t \equiv \mu(t_1, \mu(t_2, t_5))$.

Terms s and t correspond to the membranes in Fig. 9. The parts of the LTS inferred from the P Algebra that are rooted in s and t have been partially depicted in the figure.

By construction, the LTS takes into account that s and t could be inserted into a parent membrane, thus receiving objects b and c from such a parent. LTS transitions modeling a computation step in which the input received from the parent membrane is $b^p c^q$, enter state $s^{p,q}_i$, or $t^{p,q}_i$. From each state $s^{p,q}_i$, or $t^{p,q}_i$, transitions labeled $l^{p',q'}_{j,p,q}$ depart for each p' and q'. For readability, we have depicted only transitions labeled $l^{p,q}_{j,p,q}$ to represent all of them. A transition labeled $l^{p',q'}_{1,p,q}$ (resp.: $l^{p',q'}_{2,p,q}$, $l^{p',q'}_{3,p,q}$) describes a step with input $b^{p'}c^{q'}$ and output $b^p c^q$ (resp.: $b^{p+1}c^q$, $b^p c^{q+1}$).

Now, it holds that $s \approx_T t$, $s \approx_{RT} t$, and $s \approx_F t$. Relation $R = \{(s,t)\} \cup \{(s^{p,q}_2, t^{p,q}_{2,3}) \mid p,q \geq 0\} \cup \{(s^{p,q}_3, t^{p,q}_{2,3}) \mid p,q \geq 0\} \cup \{(s^{p,q}_i, t^{p,q}_i) \mid p,q \geq 0, 4 \leq i \leq 9\}$ is a simulation, whereas no simulation (and, therefore, no ready simulation) containing (t,s) can be given since neither $s^{p,q}_2$ nor $s^{p,q}_3$ are able to simulate $t^{p,q}_{2,3}$. In fact, $t^{p,q}_{2,3}$ has, among the others, both traces $l^{p,q}_{1,p,q} l^{p,q}_{1,p,q} l^{p,q}_{2,p,q}$ and $l^{p,q}_{1,p,q} l^{p,q}_{1,p,q} l^{p,q}_{3,p,q}$, whereas $s^{p,q}_2$ has not the trace $l^{p,q}_{1,p,q} l^{p,q}_{1,p,q} l^{p,q}_{3,p,q}$, and $s^{p,q}_3$ has not the trace $l^{p,q}_{1,p,q} l^{p,q}_{1,p,q} l^{p,q}_{2,p,q}$. Hence, the pair (t,s) is in $\sqsubseteq_T \setminus \sqsubseteq_S$ and in $\sqsubseteq_{RT} \setminus \sqsubseteq_{RS}$.

It remains to show that both inclusions of $\sqsubseteq_{RT}$ in $\sqsubseteq_T$ and $\sqsubseteq_{RT}$ into $\sqsubseteq_F$ are strict.

Example 14 *Let us consider the membranes in Fig. 10. The membrane 2 on the right side chooses immediately whether 1 will produce b or c at the next step, whereas membrane 2 on the left side gives to 1 the task of making the*

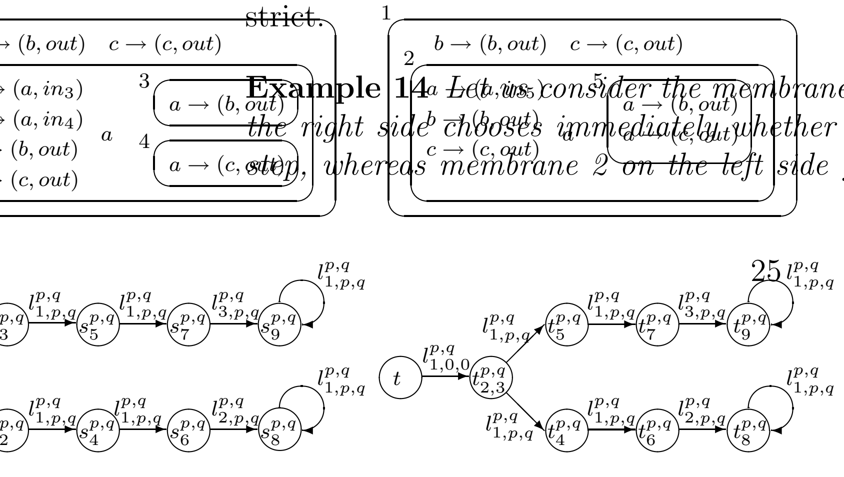

$\to (b, out)$ $c \to (c, out)$
$\to (a, in_3)$
$\to (a, in_4)$
$\to (b, out)$
$\to (c, out)$
a
3
$a \to (b, out)$
4
$a \to (c, out)$
1
$b \to (b, out)$ $c \to (c, out)$
2
$a \to (a, in_5)$
$b \to (b, out)$
$c \to (c, out)$
a
5
$a \to (b, out)$
$a \to (c, out)$
$s^{p,q}_5$ $s^{p,q}_7$ $s^{p,q}_9$ $s^{p,q}_4$ $s^{p,q}_6$ $s^{p,q}_8$
t $l^{p,q}_{1,0,0}$ $t^{p,q}_{2,3}$
$t^{p,q}_5$ $t^{p,q}_7$ $t^{p,q}_9$ $t^{p,q}_4$ $t^{p,q}_6$ $t^{p,q}_8$
$l^{p,q}_{1,p,q}$ $l^{p,q}_{2,p,q}$ $l^{p,q}_{3,p,q}$

Fig. 10. Two membrane systems showing that both inclusions $\sqsubseteq_{RT} \subseteq \sqsubseteq_F$ and $\sqsubseteq_{RT} \subseteq \sqsubseteq_T$ are strict. The labels are as follows: $l_1^{p,q,r} = (\varnothing, (1, a^p b^q c^r), \varnothing, \varnothing)$, $l_{2,u,v,w}^{p_b,p_c,q,r} = (\varnothing, (1, a^u b^v c^w), a^u b^{p_b+q+1+v} c^{p_c+r+w}, \varnothing)$, $l_{3,u,v,w}^{p_b,p_c,q,r} = (\varnothing, (1, a^u b^v c^w), a^u b^{p_b+q+v} c^{p_c+r+1+w}, \varnothing)$.

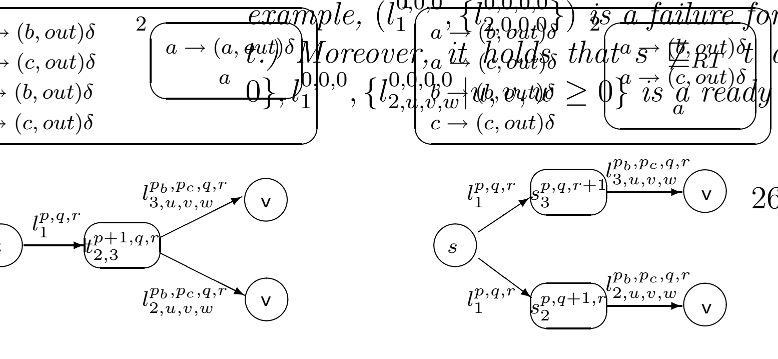
choice at next step.

Let us take the following P Algebra terms:
$s_2 \equiv [_2\, (\varnothing, a) \cup (a \to \varnothing b\delta, \varnothing) \cup (a \to \varnothing c\delta, \varnothing)\,]_2$
$t_2 \equiv [_2\, (\varnothing, a) \cup (a \to \varnothing a\delta, \varnothing)\,]_2$
$s_1 \equiv t_1 \equiv [_1\, (a \to \varnothing b \varnothing \delta, \varnothing) \cup (a \to \varnothing c \varnothing \delta, \varnothing) \cup (b \to \varnothing b \varnothing \delta, \varnothing) \cup (c \to \varnothing c \varnothing \delta, \varnothing)\,]_1$
$s \equiv \mu(s_1, s_2)$
$t \equiv \mu(t_1, t_2)$.

Terms t and s correspond to the membranes in Fig. 10. The parts of the LTS rooted in t and s have been partially depicted in the figure.

The LTS takes into account that t and s could be inserted within some parent membrane, thus receiving a, b, c from such a parent. For each $p, q, r \in \mathbb{N}$, the transition labeled $l_1^{p,q,r}$ describes the computation step performed from s, and t, for input $a^p b^q c^r$. For readability we have depicted only one transition. Term t reaches $t_{2,3}^{p+1,q,r}$, i.e. a state with objects $a^{p+1} b^q c^r$. Note that one of the objects a has been produced by the membrane 2. Term s reaches either $s_2^{p,q+1,r}$, i.e. a state with objects $a^p b^{q+1} c^r$, where one object b has been produced by the membrane 2, or $s_3^{p,q,r+1}$, i.e. a state with objects $a^p b^q c^{r+1}$, where one object c has been produced by the membrane 2. Label $l_{2,u,v,w}^{p_b,p_c,q,r}$ (resp. $l_{3,u,v,w}^{p_b,p_c,q,r}$) describes a computation step for input $a^u b^v c^w$ and output $a^u b^{p_b+q+1+v} c^{p_c+r+w}$ (resp. $a^u b^{p_b+q+v} c^{p_c+r+1+w}$), where p_b and p_c are values such such that $p_b + p_c = p$, and the output contains the input due to dissolution.

It holds that $s \approx_T t$ and $t \sqsubseteq_F s$ (Note, on the contrary, that $s \not\sqsubseteq_F t$. For example, $(l_1^{0,0,0}, \{l_{2,0,0,0}^{0,0,0,0}\})$ is a failure for state s but it is not a failure for state t.) Moreover, it holds that $s \not\sqsubseteq_{RT} t$ and $t \not\sqsubseteq_{RT} s$. In fact, $\{l_1^{p,q,r} | p,q,r \geq 0\}, l_1^{0,0,0}, \{l_{2,u,v,w}^{0,0,0,0} | u,v,w \geq 0\}$ is a ready trace of s but it is not a ready trace of



t. Viceversa, $\{l_1^{p,q,r} | p,q,r \geq 0\}, l_1^{0,0,0}, \{l_{2,u,v,w}^{0,0,0,0}, l_{3,u,v,w}^{0,0,0,0}\}$ is a ready trace of t but not of s. Hence, both (s,t) and (t,s) are in $\sqsubseteq_T \setminus \sqsubseteq_{RT}$, and (t,s) is in $\sqsubseteq_F \setminus \sqsubseteq_{RT}$.

5 Priorities

We can deal with priority by adding to evolution rules information on the evolution rules having higher priority. Priorities are usually represented as a binary relation on evolution rules. However, for the sake of compositionality, we would like that each evolution rule contains some information about its priority relationships with respect to other rules. In particular, we would like that each evolution rule contains some information about the applicability of rules with higher priority.

To model that a rule $u \to v_h v_o \{v_{l_i}\}$ has lower priority than all evolution rules in a set $\{u_j \to (v_h^j, here)(v_o^j, out)(v_1^j, in_{l_1^j}), \ldots (v_{n_j}^j, in_{l_{n_j}^j}) \,|\, j \in J\}$, our choice is to represent $u \to v_h v_o \{v_{l_i}\}$ with the constant $(\{(u_j, M_j)\}\, u \to v_h v_o \{v_{l_i}\}, \varnothing)$, where $M_j = \{l_1^j, \ldots, l_{n_j}^j\}$. The interpretation is that $u \to v_h v_o \{v_{l_i}\}$ cannot be applied if there exists some j such that all objects in u_j are available and all labels in M_j are labels of child membranes. In fact, in such a case, a rule with higher priority is applicable.

In the following, a pair in $V^* \times 2^{\mathbb{N}}$ as (u_j, M_j) is called a *priority pair.*

Hence, we assume that non–dissolving evolution rules and dissolving evolution rules have the following forms:

$$(u_1, M_1) \ldots (u_m, M_m)\; u \to (v_h, here)(v_o, out)(v_1, in_{l_1}) \ldots (v_n, in_{l_n})$$

$$(u_1, M_1) \ldots (u_m, M_m)\; u \to (v_h, here)(v_o, out)(v_1, in_{l_1}) \ldots (v_n, in_{l_n})\delta$$

We denote them with $\{(u_i, M_i)\}\, u \to v_h v_o \{v_{l_i}\}$ and $\{(u_i, M_i)\}\, u \to v_h v_o \{v_{l_i}\}\delta$, respectively.

The Priority P Algebra (PP Algebra) is defined analogously to the P Algebra.

Definition 15 (PP Algebra) *The abstract syntax of* membrane contents c, membranes m, *and* membrane systems ms *is given by the following grammar, where l ranges over $\mathbb{N}$ and a over V:*

$$\begin{aligned} c \;::=&\; (\varnothing, \varnothing) \;\Big|\; (\{(u_i, M_i)\}\, u \to v_h v_o \{v_{l_i}\}, \varnothing) \;\Big| \\ &\; (\{(u_i, M_i)\}\, u \to v_h v_o \{v_{l_i}\}\delta, \varnothing) \;\Big|\; (\varnothing, a) \;\Big|\; c \cup c \end{aligned}$$

$$m \;::=\; [_l c]_l$$

$$ms \ ::= m \ \Big| \ ms \mid ms \ \Big| \ \mu(m, ms) \ \Big| \ \mathsf{v}$$

The LTS associated with PP Algebra has labels carrying information on priority. Labels can be of the following forms:

- $(u, U, v, v', M, I, O^{\uparrow}, O^{\downarrow}, u_P, M_P, A)$, describing a computation step performed by a membrane content c. The new components u_P, M_P, and A have the following meaning:
  - The pair (u_P, M_P) is a priority pair that explains the non application of some evolution rules in c with the triggering of other evolution rules in c having higher priority. Actually, we assume that all objects in u_P are available, and that all labels in M_P are labels of child membranes to which objects can be delivered. When operation $[_l\text{-}]_l$ is applied to c, and, subsequently, operation μ is applied to $([_l c]_l, ms)$, for any membrane system ms and membrane label l, we have to check that our assumption is correct. More precisely, we require that u_P is contained in vv', and that each label in M_P is a label of some membrane in ms.
  - A is a set of priority pairs such that each (u, M) in A would preempt at least one of the applied evolution rules in c with the triggering of another evolution rule in c having higher priority. Actually, for each $(u, M) \in A$, we assume that either some object in u is not available, or some label in M is not a label of any child membrane. When operation $[_l\text{-}]_l$ is applied to c, and, subsequently, operation μ is applied to $([_l c]_l, ms)$, for any membrane system ms and membrane label l, we have to check that our assumption is correct. More precisely, we require that either u is not contained in vv', or that some label in M is not a label of any membrane in ms.
- $(M, \mathcal{I}, O^{\uparrow}, O^{\downarrow}, M_P, A)$, describing a computation step performed by a membrane system ms. The new components M_P and A have the same meaning as in the previous case.

Rules in Fig. 11 extend those in Fig. 2 having the same name with additional information for handling priorities. In what follows we describe only the additional information.

Let us consider the rule $(mc1_n)$. The label shows that no evolution rule with higher priority is applicable, namely for each i either some object in u_i is not available, or some label in M_i is not a label of any child membrane. Hence, if $n \geq 1$ then n occurrences of $\{(u_i, M_i)\}\, u \to v_h v_o \{v_{l_i}\}$ are applied, whereas if $n = 0$ the label shows that $\{(u_i, M_i)\}\, u \to v_h v_o \{v_{l_i}\}$ is not applicable since the multiset of objects u is not available. The case in which $\{(u_i, M_i)\}\, u \to v_h v_o \{v_{l_i}\}$ is not applied since the evolution rule with higher priority enabled by the objects u_i and delivering objects to the child membranes in M_i is applicable, is described by the new rule $(mc1'_0)$. In rules $(mc2_n)$, $(mc3)$, and in the new rule $(mc3')$, priorities are handled as in rules $(mc1_n)$, $(mc1_0)$,

$$\frac{I \in V^* \qquad n \in \mathbb{N}}{(\{(u_i, M_i)\}\, u \to v_h v_o\{v_{l_i}\}, \varnothing) \xrightarrow[u^n,\{u\},\varnothing,\varnothing]{\varnothing,I,v_o^n,\{(l_i,v_{l_i}^n)\},\varnothing,\varnothing,\{(u_i,M_i)\}} (\{(u_i, M_i)\}\, u \to v_h v_o\{v_{l_i}\}, I v_h^n)} \quad (mc1_n)$$

$$\frac{I \in V^* \qquad (u_j, M_j) \in \{(u_i, M_i)\}}{(\{(u_i, M_i)\}\, u \to v_h v_o\{v_{l_i}\}, \varnothing) \xrightarrow[\varnothing,\varnothing,\varnothing,\varnothing]{\varnothing,I,\varnothing,\varnothing,u_j,M_j,\varnothing} (\{(u_i, M_i)\}\, u \to v_h v_o\{v_{l_i}\}, I)} \quad (mc1'_0)$$

$$\frac{I \in V^* \qquad n \in \mathbb{N} \qquad n > 0}{(\{(u_i, M_i)\}\, u \to v_h v_o\{v_{l_i}\}\delta, \varnothing) \xrightarrow[u^n,\{u\},\varnothing,\varnothing]{\varnothing,I,I v_o^n v_h^n \delta,\{(l_i,v_{l_i}^n)\},\varnothing,\varnothing,\{(u_i,M_i)\}} \mathsf{v}} \quad (mc2_n)$$

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

$$\frac{I \in V^* \qquad (u_j, M_j) \in \{(u_i, M_i)\}}{(\{(u_i, M_i)\}\, u \to v_h v_o\{v_{l_i}\}\delta, \varnothing) \xrightarrow[\varnothing,\varnothing,\varnothing,\varnothing]{\varnothing,I,\varnothing,\varnothing,u_j,M_j,\varnothing} (\{(u_i, M_i)\}\, u \to v_h v_o\{v_{l_i}\}\delta, I)} \quad (mc3')$$

$$\frac{I \in V^* \qquad M \subseteq \mathsf{Labels}(\{v_{l_i}\}) \qquad M \neq \varnothing}{(\{(u_i, M_i)\}\, u \to v_h v_o\{v_{l_i}\}, \varnothing) \xrightarrow[\varnothing,\varnothing,\varnothing,\varnothing]{M,I,\varnothing,\varnothing,\varnothing,\varnothing,\{(u_i,M_i)\}} (\{(u_i, M_i)\}\, u \to v_h v_o\{v_{l_i}\}, I)} \quad (mc4)$$

$$\frac{I \in V^* \qquad M \subseteq \mathsf{Labels}(\{v_{l_i}\}) \qquad M \neq \varnothing}{(\{(u_i, M_i)\}\, u \to v_h v_o\{v_{l_i}\}\delta, \varnothing) \xrightarrow[\varnothing,\varnothing,\varnothing,\varnothing]{M,I,\varnothing,\varnothing,\varnothing,\varnothing,\{(u_i,M_i)\}} (\{(u_i, M_i)\}\, u \to v_h v_o\{v_{l_i}\}\delta, I)} \quad (mc5)$$

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

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

Fig. 11. Rules for membrane contents.

and $(mc1'_0)$, respectively. In rules $(mc4)$ and $(mc5)$ the label shows that no evolution rule with higher priority is applicable. In rules $(mc6)$, $(mc7)$, and $(mc8)$ priority plays no role.

Given two multisets of objects u_1 and u_2, let $u_1 \uplus u_2$ denote the multiset $(u_1 \cup u_2) \setminus (u_1 \cap u_2)$. If u_1 (resp. u_2) is a multiset of objects that are assumed to be available so that no evolution rule in a set R_1 (resp. R_2) is applicable, then $u_1 \uplus u_2$ is the least multiset of objects that should be available so that no evolution rule in $R_1 \cup R_2$ is applicable. This operation is exploited in the rules in Fig. 12, which extend those in Fig. 3 with handling of priorities. In all these rules, all priority pairs (u, M) in $A_1 A_2$ are such that either some object in u is assumed to be not available, or some label in M is assumed not to be a label of any child membrane. Since all objects in $u_{P_1} \uplus u_{P_2}$ are assumed to be available, and all labels in M_{P_1} and M_{P_2} are assumed to be labels of child membranes, we require that either $u \not\subseteq u_{P_1} \uplus u_{P_2}$ or $M \not\subseteq M_{P_1} M_{P_2}$. Moreover, since all labels in $M_1 M_2$ are assumed not to be labels of child membranes, we

$$\frac{\begin{array}{c} M_1M_2 \cap M_{P_1}M_{P_2} = \varnothing \quad \forall (u,M) \in A_1A_2.(u \not\subseteq u_{P_1} \uplus u_{P_2} \vee M \not\subseteq M_{P_1}M_{P_2}) \\ M_1M_2 \cap \mathsf{Labels}(O_1^{\downarrow} \cup_{\mathbb{N}} O_2^{\downarrow}) = \varnothing \qquad v'_1v'_2 \nvdash U_1 \oplus U_2 \qquad \delta \notin O_1^{\uparrow}O_2^{\uparrow} \\ x_1 \xrightarrow[u_1,U_1,v_1,v'_1]{M_1,I_1,O_1^{\uparrow},O_1^{\downarrow},u_{P_1},M_{P_1},A_1} y_1 \qquad x_2 \xrightarrow[u_2,U_2,v_2,v'_2]{M_2,I_2,O_2^{\uparrow},O_2^{\downarrow},u_{P_2},M_{P_2},A_2} y_2 \end{array}}{x_1 \cup x_2 \xrightarrow[u_1u_2,U_1\oplus U_2,v_1v_2,v'_1v'_2]{M_1M_2,I_1I_2,O_1^{\uparrow}O_2^{\uparrow},O_1^{\downarrow}\cup_{\mathbb{N}}O_2^{\downarrow},u_{P_1}\uplus u_{P_2},M_{P_1}M_{P_2},A_1A_2} y_1 \cup y_2} \quad (u1)$$

$$\frac{\begin{array}{c} M_1M_2 \cap M_{P_1}M_{P_2} = \varnothing \quad \forall (u,M) \in A_1A_2.(u \not\subseteq u_{P_1} \uplus u_{P_2} \vee M \not\subseteq M_{P_1}M_{P_2}) \\ M_1M_2 \cap \mathsf{Labels}(O_1^{\downarrow} \cup_{\mathbb{N}} O_2^{\downarrow}) = \varnothing \qquad v'_1v'_2 \nvdash U_1 \oplus U_2 \qquad \delta \in O_1^{\uparrow} \quad \delta \notin O_2^{\uparrow} \\ x_1 \xrightarrow[u_1,U_1,v_1,v'_1]{M_1,I_1,O_1^{\uparrow},O_1^{\downarrow},u_{P_1},M_{P_1},A_1} y_1 \qquad x_2 \xrightarrow[u_2,U_2,v_2,v'_2]{M_2,I_2,O_2^{\uparrow},O_2^{\downarrow},u_{P_2},M_{P_2},A_2} y_2 \end{array}}{x_1 \cup x_2 \xrightarrow[u_1u_2,U_1\oplus U_2,v_1v_2,v'_1v'_2]{M_1M_2,I_1I_2,O_1^{\uparrow}O_2^{\uparrow}\mathsf{Objects}(y_2),O_1^{\downarrow}\cup_{\mathbb{N}}O_2^{\downarrow},u_{P_1}\uplus u_{P_2},M_{P_1}M_{P_2},A_1A_2} \mathsf{v}} \quad (u2)$$

$$\frac{\begin{array}{c} M_1M_2 \cap M_{P_1}M_{P_2} = \varnothing \quad \forall (u,M) \in A_1A_2.(u \not\subseteq u_{P_1} \uplus u_{P_2} \vee M \not\subseteq M_{P_1}M_{P_2}) \\ M_1M_2 \cap \mathsf{Labels}(O_1^{\downarrow} \cup_{\mathbb{N}} O_2^{\downarrow}) = \varnothing \qquad v'_1v'_2 \nvdash U_1 \oplus U_2 \qquad \delta \in O_1^{\uparrow} \cap O_2^{\uparrow} \\ x_1 \xrightarrow[u_1,U_1,v_1,v'_1]{M_1,I_1,O_1^{\uparrow},O_1^{\downarrow},u_{P_1},M_{P_1},A_1} y_1 \qquad x_2 \xrightarrow[u_2,U_2,v_2,v'_2]{M_2,I_2,O_2^{\uparrow},O_2^{\downarrow},u_{P_2},M_{P_2},A_2} y_2 \end{array}}{x_1 \cup x_2 \xrightarrow[u_1u_2,U_1\uplus U_2,v_1v_2,v'_1v'_2]{M_1M_2,I_1I_2,O_1^{\uparrow}O_2^{\uparrow},O_1^{\downarrow}\cup_{\mathbb{N}}O_2^{\downarrow},u_{P_1}\uplus u_{P_2},M_{P_1}M_{P_2},A_1A_2} \mathsf{v}} \quad (u3)$$

Fig. 12. Rules for union of membrane contents.

require that $M_1M_2 \cap M_{P_1}M_{P_2} = \varnothing$.

Let us consider the rules in Fig. 13, which extend those in Fig. 5 with handling of priority. Let us take rules $(m1)$ and $(m2)$. Since all objects in u_P are assumed to be available, we require that $u_P \subseteq uv'$. Since for each pair $(u', M') \in A$ it is assumed that either some object in u' is not available, or some label in M' is not a label of any child membrane, we require that if $u' \subseteq uv'$, namely if the objects in u' are actually available, then M' is not empty. In such a case, if operation μ is not applied to $([_l x]_l, ms)$, for any ms, then we are sure that l has no child with label in M', and, if μ is applied to $([_l x]_l, ms)$, for some ms, then we shall check that some label in M' is not a label of any child in ms. To this purpose, it suffices to carry only information on M' in the label of the conclusion. Let us consider rules $(jux1)$, $(jux2)$, and $(jux3)$. Since Def. 15 ensures that $x_1 \mid x_2$ cannot appear as the first argument of operation μ, namely x_1 and x_2 cannot have any common child, we are sure that the set of children of x_1 and x_2 cannot furtherly change. Hence, we can assume that we have already checked that all membrane labels that are required to be labels of children of x_1 and x_2 actually exist, as it appears from the fifth component of the label of the premises, which are empty. For the same reason, we are sure

$$\frac{x \xrightarrow[u,U,u,v']{M,I,O^{\uparrow},O^{\downarrow},u_P,M_P,A} y \quad \delta \notin O^{\uparrow} \quad u_P \subseteq uv' \quad \forall (u',M') \in A.(u' \subseteq uv' \implies M' \neq \varnothing)}{[_l x]_l \xrightarrow{M,\{(l,I)\},O^{\uparrow},O^{\downarrow},M_P,\{(\varnothing,M') \mid \exists u'.(u',M') \in A \wedge u' \subseteq uv'\}} [_l y]_l} \quad (m1)$$

$$\frac{x \xrightarrow[u,U,u,v']{M,I,O^{\uparrow},O^{\downarrow},u_P,M_P,A} y \quad \delta \in O^{\uparrow} \quad u_P \subseteq uv' \quad \forall (u',M') \in A.(u' \subseteq uv' \implies M' \neq \varnothing)}{[_l x]_l \xrightarrow{M,\{(l,I)\},O^{\uparrow},O^{\downarrow},M_P,\{(\varnothing,M') \mid \exists u'.(u',M') \in A \wedge u' \subseteq uv'\}} \mathsf{v}} \quad (m2)$$

$$\frac{x_1 \xrightarrow{M_1,\mathcal{I}_1,O_1^{\uparrow},\varnothing,\varnothing,A_1} y_1 \quad x_2 \xrightarrow{M_2,\mathcal{I}_2,O_2^{\uparrow},\varnothing,\varnothing,A_2} y_2 \quad \delta \notin O_1^{\uparrow} O_2^{\uparrow}}{x_1|x_2 \xrightarrow{\varnothing,\mathcal{I}_1\mathcal{I}_2,O_1^{\uparrow}O_2^{\uparrow},\varnothing,\varnothing,\varnothing} y_1|y_2} \quad (jux1)$$

$$\frac{x_1 \xrightarrow{M_1,\mathcal{I}_1,O_1^{\uparrow},\varnothing,\varnothing,A_1} y_1 \quad x_2 \xrightarrow{M_2,\mathcal{I}_2,O_2^{\uparrow},\varnothing,\varnothing,A_2} y_2 \quad \delta \in O_1^{\uparrow}, \delta \notin O_2^{\uparrow}}{x_1|x_2 \xrightarrow{\varnothing,\mathcal{I}_1\mathcal{I}_2,(O_1^{\uparrow}O_2^{\uparrow})-\delta,\varnothing,\varnothing,\varnothing} y_2} \quad (jux2)$$

$$\frac{x_1 \xrightarrow{M_1,\mathcal{I}_1,O_1^{\uparrow},\varnothing,\varnothing,A_1} y_1 \quad x_2 \xrightarrow{M_2,\mathcal{I}_2,O_2^{\uparrow},\varnothing,\varnothing,A_2} y_2 \quad \delta \in O_1^{\uparrow} \cap O_2^{\uparrow}}{x_1|x_2 \xrightarrow{\varnothing,\mathcal{I}_1\mathcal{I}_2,(O_1^{\uparrow}O_2^{\uparrow}),\varnothing,\varnothing,\varnothing} \mathsf{v}} \quad (jux3)$$

Fig. 13. Rules for single membranes and juxtaposition of membranes.

that the labels we have assumed not to be labels of children of x_1 and x_2 and that are carried by A_1 and A_2 actually are not labels of any child of x_1 and x_2, and we can set to $\varnothing$ the last component of the label of the conclusion.

Let us consider the rules in Fig. 14, which extend those in Fig. 6 with handling of priority. In all rules, since all membrane labels in M_P are assumed to be labels of children of x_1, we require that $M_P \subseteq \mathsf{Labels}(\mathcal{I}_2)$. The rules in Fig. 13 ensure that, for each $(u, M) \in A$, it holds that $u = \varnothing$. Hence, we have to check that at least one membrane label in M is not a label of any child of x_1. Actually, we check that $M \not\subseteq \mathsf{Labels}(\mathcal{I}_2)$. Since Def. 15 ensures that $\mu(x_1, x_2)$ cannot appear as the first argument of operation μ, namely x_1 and x_2 cannot have any common child, we are sure that the set of children of x_1 and x_2 cannot furtherly change. Hence, we can assume that we have already checked that all membrane labels that are required to be labels of children of x_2 actually exist, and that no membrane label that is required not to be a label of any child of x_2 actually exists, as it appears from the fifth and the sixth component of the label of the transition performed by x_2 in the premise, which are empty sets.

5.1 *Properties of the Semantics*

First of all let us note that also the SOS rules of the semantics of the PP Algebra respect the de Simone format, thus implying that the precongruence

$$\frac{M_1 \cap \mathsf{Labels}(\mathcal{I}_2) = \varnothing \quad O_1^{\downarrow} \bumpeq \mathcal{I}_2 \quad O_2^{\uparrow} \subseteq I_1 \quad M_P \subseteq \mathsf{Labels}(\mathcal{I}_2) \quad \forall (u, M) \in A . M \nsubseteq \mathsf{Labels}(\mathcal{I}_2) \\ x_1 \xrightarrow{M_1, \{(l_1, I_1)\}, O_1^{\uparrow}, O_1^{\downarrow}, M_P, A} y_1 \quad x_2 \xrightarrow{M_2, \mathcal{I}_2, O_2^{\uparrow}, \varnothing, \varnothing, \varnothing} y_2 \quad \delta \notin O_1^{\uparrow} O_2^{\uparrow}}{\mu(x_1, x_2) \xrightarrow{\varnothing, (l_1, I_1 \setminus O_2^{\uparrow}), O_1^{\uparrow}, \varnothing, \varnothing, \varnothing} \mu(y_1, y_2)} \tag{$h1$}$$

$$\frac{M_1 \cap \mathsf{Labels}(\mathcal{I}_2) = \varnothing \quad O_1^{\downarrow} \bumpeq \mathcal{I}_2 \quad O_2^{\uparrow} \subseteq I_1 \quad M_P \subseteq \mathsf{Labels}(\mathcal{I}_2) \quad \forall (u, M) \in A . M \nsubseteq \mathsf{Labels}(\mathcal{I}_2) \\ x_1 \xrightarrow{M_1, \{(l_1, I_1)\}, O_1^{\uparrow}, O_1^{\downarrow}, M_P, A} y_1 \quad x_2 \xrightarrow{M_2, \mathcal{I}_2, O_2^{\uparrow}, \varnothing, \varnothing, \varnothing} y_2 \quad \delta \in O_1^{\uparrow} \quad \delta \notin O_2^{\uparrow}}{\mu(x_1, x_2) \xrightarrow{\varnothing, \{(l_1, I_1 \setminus O_2^{\uparrow})\}, O_1^{\uparrow} - \delta, \varnothing, \varnothing, \varnothing} y_2} \tag{$h2$}$$

$$\frac{M_1 \cap \mathsf{Labels}(\mathcal{I}_2) = \varnothing \quad O_1^{\downarrow} \bumpeq \mathcal{I}_2 \quad O_2^{\uparrow} \subseteq I_1 \quad M_P \subseteq \mathsf{Labels}(\mathcal{I}_2) \quad \forall (u, M) \in A . M \nsubseteq \mathsf{Labels}(\mathcal{I}_2) \\ x_1 \xrightarrow{M_1, \{(l_1, I_1)\}, O_1^{\uparrow}, O_1^{\downarrow}, M_P, A} y_1 \quad x_2 \xrightarrow{M_2, \mathcal{I}_2, O_2^{\uparrow}, \varnothing, \varnothing, \varnothing} y_2 \quad \delta \notin O_1^{\uparrow} \quad \delta \in O_2^{\uparrow}}{\mu(x_1, x_2) \xrightarrow{\varnothing, \{(l_1, I_1 \setminus O_2^{\uparrow})\}, O_1^{\uparrow}, \varnothing, \varnothing, \varnothing} y_1} \tag{$h3$}$$

$$\frac{M_1 \cap \mathsf{Labels}(\mathcal{I}_2) = \varnothing \quad O_1^{\downarrow} \bumpeq \mathcal{I}_2 \quad O_2^{\uparrow} \subseteq I_1 \quad M_P \subseteq \mathsf{Labels}(\mathcal{I}_2) \quad \forall (u, M) \in A . M \nsubseteq \mathsf{Labels}(\mathcal{I}_2) \\ x_1 \xrightarrow{M_1, \{(l_1, I_1)\}, O_1^{\uparrow}, O_1^{\downarrow}, M_P, A} y_1 \quad x_2 \xrightarrow{M_2, \mathcal{I}_2, O_2^{\uparrow}, \varnothing, \varnothing, \varnothing} y_2 \quad \delta \in O_1^{\uparrow} \cap O_2^{\uparrow}}{\mu(x_1, x_2) \xrightarrow{\varnothing, \{(l_1, I_1 \setminus O_2^{\uparrow})\}, O_1^{\uparrow}, \varnothing, \varnothing, \varnothing} \mathsf{v}} \tag{$h4$}$$

Fig. 14. Rules for hierarchy of membranes.

results stated in Thm. 10 are still valid. Moreover, the examples given in Sect. 4 to show that the inclusions in Fig. 8 are strict in the LTS inferred from the P Algebra are valid for the LTS inferred from the PP Algebra as well.

Theorem 16 *All preorders in Def. 7 are precongruences.*

PROOF. The same as that of Theorem 10. □

Corollary 17 *The kernels of all preorders in Def. 7 are congruences.*

Our aim is now to show that the semantics of the PP Algebra does a correct handling of priorities. Showing that the semantics reflects maximal parallelism can be done as in the case of the P Algebra.

In the derivation of a transition of a term of the form (R, u'), or $[_l(R, u')]_l$, or $\mu([_l(R, u')]_l, x')$, we say that a non dissolving evolution rule (resp. dissolving evolution rule) r in R is *assumed to be enabled* when a semantic rule among $(mc1_n)$ and $(mc4)$ (resp. $(mc2_n)$, $(mc3)$, and $(mc5)$) is applied to infer the transition of $(r, \varnothing)$ exploited to infer the transition of the whole term. Notice that these semantic rules are precisely the rules for $(r, \varnothing)$ assuming that no priority pair of r preempts r.

Definition 18 *Let $r = \{(u_i, M_i)\}\, u \to v_h v_o\{v_{l_i}\}$ (resp. $r = \{(u_i, M_i)\}\, u \to v_h v_o\{v_{l_i}\}\delta$) be an evolution rule and $(\mathcal{R}, u') = (\mathcal{R}', u') \cup (r, \varnothing)$ be a membrane content. We say that r is* assumed to be enabled *in the derivation of a transition $(\mathcal{R}, u') \xrightarrow[u'',U,v,v']{M,I,O^{\uparrow},O^{\downarrow},u_P,M_P,A} x$ inferred by applying the semantic rule either $(u1)$ or $(u2)$ (resp. $(u2)$ or $(u3)$), with premises $(\mathcal{R}', u') \xrightarrow[u_1,U_1,v_1,v_1']{M_1,I_1,O_1^{\uparrow},O_1^{\downarrow},u_{P_1},M_{P_1},A_1} y_1$ and $(r, \varnothing) \xrightarrow[u_2,U_2,v_2,v_2']{M_2,I_2,O_2^{\uparrow},O_2^{\downarrow},u_{P_2},M_{P_2},A_2} y_2$, if and only if this latter premise is inferred by applying the rule $(mc1_n)$ or $(mc4)$ (resp. $(mc2_n)$, or $(mc3)$, or $(mc5)$).*

Moreover, r is assumed to be enabled in the derivation of a transition performed by the membrane $[_l(\mathcal{R}, u')]_l$ inferred by applying the rule $(m1)$ or $(m2)$ with a transition of $(\mathcal{R}, u')$ in which r is assumed to be enabled as premise

Finally, r is assumed to be enabled in the derivation of a transition performed by the hierarchy $\mu([_l(\mathcal{R}, u')]_l, x')$ inferred by applying a rule in $(h1)$–$(h4)$ with a transition of $[_l(\mathcal{R}, u')]_l$ in which r is assumed to be enabled as premise.

To show that priorities are handled correctly, we can show that the transitions in which an evolution rule r is assumed to be enabled can be derived only if either some of the objects or some of the child membranes needed to trigger the rules with higher priority are not present, namely no priority pair of r preempts r. To this aim, we prove two theorems. The first theorem states that, if r is assumed to be enabled in the derivation of a transition performed by a membrane $[_l(R, u')]_l$ with r in R, then, for each of the priority pairs of r in which no membrane label appears, at least one object among those in the priority pair is not in u'. Notice that in this case we do not consider priority pairs in which at least one membrane label l' appears. The reason is that, since l has no child, we are sure that no membrane labeled l' can be a child of l. The second theorem states that, if r has some membrane label l' in its priority pairs and is assumed to be enabled in the derivation of a transition performed by the hierarchy $\mu([_l(R, u')]_l, x')$ with r in R, then no membrane in x' has label l'. Notice that in this case we do not consider objects in priority pairs since they play no role in hierarchy.

To prove the first theorem we need the following lemma, which states that if a rule r is assumed to be enabled in the derivation of a transition of the membrane content $(\mathcal{R}, u')$, then the label of the transition keeps track of all priority pairs that would preempt r.

Lemma 19 *If a rule $\{(u_i, M_i)\}\, u \to v_h v_o\{v_{l_i}\}$ or $\{(u_i, M_i)\}\, u \to v_h v_o\{v_{l_i}\}\delta$ is* *assumed to be enabled in the derivation of a transition* $(\mathcal{R}, u') \xrightarrow[u'',U,v,v']{M,I,O^{\uparrow},O^{\downarrow},u_P,M_P,A} x$*, then* $\{(u_i, M_i)\} \subseteq A$*.*

PROOF. By induction on the derivation of the transition. The base cases are when one of the semantic rules in Fig. 11 is applied. The property $\{(u_i, M_i)\} \subseteq A$ is satisfied in the cases of $(mc1_n)$, $(mc2_n)$, $(mc3)$, $(mc4)$ and $(mc5)$, while in the other cases either there is no evolution rule, or the evolution rule is not assumed to be enabled. The induction cases are when one of the rules in Fig. 12 is applied, and in this case the induction hypothesis can be applied trivially. □

Theorem 20 *If an evolution rule* $r = \{(u_i, M_i)\}\, u \to v_h v_o \{v_{l_i}\}$ *or* $r = \{(u_i, M_i)\}\, u \to v_h v_o \{v_{l_i}\}\delta$ *is assumed to be enabled in the derivation of a transition* $[_l(\mathcal{R}, u')]_l \xrightarrow{M,\mathcal{I},O^{\uparrow},O^{\downarrow},M_P,A} x$*, then, for all* $(v, \varnothing) \in \{(u_i, M_i)\}$*, it holds that* $v \not\subseteq u'$*.*

PROOF. By Def. 18 we know that transition $[_l(\mathcal{R}, u')]_l \xrightarrow{M,\mathcal{I},O^{\uparrow},O^{\downarrow},M_P,A} x$ is derived by applying semantic rule either $(m1)$ or $(m2)$, and by using as premise a transition $(\mathcal{R}, u') \xrightarrow[u'',U,u'',v']{M,I,O^{\uparrow},O^{\downarrow},u_P,M_P,A'} y$ in which r is assumed to be enabled. By Lemma 19 we know that $\{(u_i, M_i)\} \subseteq A'$. Finally, by definition of $(m1)$ and $(m2)$, we have that, for all $(v, \varnothing) \in \{(u_i, M_i)\}$, it holds that $v \not\subseteq u''v' = u'$. □

To prove the second theorem we need the following lemma, which states that if a rule r is assumed to be enabled in the derivation of a transition of the membrane $[_l(\mathcal{R}, u')]_l$, then the label of the transition keeps track of the membrane labels that should not appear in any membrane system put inside $[_l(\mathcal{R}, u')]_l$.

Lemma 21 *If a rule* $\{(u_i, M_i)\}\, u \to v_h v_o \{v_{l_i}\}$ *or* $\{(u_i, M_i)\}\, u \to v_h v_o \{v_{l_i}\}\delta$ *is assumed to be enabled in the derivation of a transition* $[_l(\mathcal{R}, u')]_l \xrightarrow{M,I,O^{\uparrow},O^{\downarrow},M_P,A} x$*, then, for each* $(v, M') \in \{(u_i, M_i)\}$ *such that* $v \subseteq u'$*, it holds that* $(\varnothing, M') \in A$*.*

PROOF. Similar to the proof of Theorem 20. □

Let $\mathsf{Lab}(x_1 \mid \ldots \mid x_n)$, where x_i is either $[_{l_i}(\mathcal{R}, u)]_{l_i}$ or $\mu([_{l_i}(\mathcal{R}, u)]_{l_i}, y_i)$, be the set of labels of the juxtaposition of membranes, namely the set $\{l_1, \ldots, l_n\}$.

Theorem 22 *If an evolution rule* $r = \{(u_i, M_i)\}\, u \rightarrow v_h v_o \{v_{l_i}\}$ *or* $r = \{(u_i, M_i)\}\, u \rightarrow v_h v_o \{v_{l_i}\}\delta$ *is assumed to be enabled in the derivation of a transition* $\mu([_l(\mathcal{R}, u')]_l, x) \xrightarrow{M,\mathcal{I},O^{\uparrow},O^{\downarrow},M_P,A} y$*, then, for all* $(v, M') \in \{(u_i, M_i)\}$ *such that* $v \subseteq u'$*, it holds that* $M' \not\subseteq \mathsf{Lab}(x)$*.*

PROOF. By Def. 18 we know that the transition $\mu([_l(\mathcal{R}, u')]_l, x) \xrightarrow{M,\mathcal{I},O^{\uparrow},O^{\downarrow},M_P,A}$ y is derived by applying a semantic rule among $(h1)$–$(h4)$, and by using as a premise a transition $[_l(\mathcal{R}, u')]_l \xrightarrow{M_1,\mathcal{I}_1,O_1^{\uparrow},O_1^{\downarrow},M_P',A'} y$ in which r is assumed to be enabled. By Lemma 21 we know that for all $(v, M') \in \{(u_i, M_i)\}$ such that $v \subseteq u'$, it holds that $(\varnothing, M') \in A'$. Finally, the other premise of the rule $(h1) - -(h4)$ has the form $x \xrightarrow{M_2,\mathcal{I}_2,O_2^{\uparrow},\varnothing,\varnothing,\varnothing} x'$, and the constraints of the rule implies that $M' \not\subseteq \mathsf{Labels}(\mathcal{I}_2) = \mathsf{Lab}(x)$. □

6 Future Work

It would be worth developing axiomatic semantics characterizing equivalent P Systems. Namely, given any equivalence relation, we should provide a set of syntactical transformations between terms being correct and complete w.r.t. the equivalence considered. This has application in program transformation and proof by rewriting.

In the field of classic process calculi, axiomatic theories are well established since the eighties [6,15], and, for languages in suitable classes, algorithms have been developed to obtain axiomatizations in a syntax-driven way (as examples, see [1,2]). A central idea in the mentioned papers is that concurrency can be simulated by interleaving, namely the concurrent execution of two actions, say a and b, is simulated by the nondeterministic choice between performing a and subsequently b, or conversely, b and subsequently a. This is possible because processes running in parallel run at different rates, and one cannot predict the relative temporal order between their actions. Concurrent processes can be reduced to nondeterministic choices of sequential processes that are called *head normal forms*.

Let us take now the P Algebra terms $(a \rightarrow \varnothing b \varnothing, \varnothing)$ and $(c \rightarrow \varnothing d \varnothing, \varnothing)$. Their union w.r.t. operation $(\_ \cup \_)$ can be viewed as a parallel composition of two terms, each representing an evolution rule. Since these rules are in the same membrane, they are applied with maximal parallelism, so that n occurrences of a and m occurrences of c imply n application of the first rule and m application of the second rule, so that n occurrences of b and m occurrences of d are sent to the outer membrane. Now, the behavior of $(a \rightarrow \varnothing b \varnothing, \varnothing) \cup (c \rightarrow \varnothing d \varnothing, \varnothing)$

cannot be equivalent to the behavior of any membrane content having only one evolution rule. This implies that parallelism cannot be reduced, at least at membrane content level. As a consequence, the work by [6,15] based on reduction of parallelism and on head normal forms cannot be adapted to P Systems setting in a trivial way. Similar problems arise in axiomatizations for synchronous languages, where parallelism cannot be reduced since processes running in parallel are perfectly synchronized and proceed at the same rate [20,21].

Acknowledgments

This research has been partially supported by MiUR PRIN 2005 Project "Analisi di sistemi di Riduzione mediante sistemi di Transizione (ART)" and MiUR PRIN 2006 Project "Biologically Inspired Systems and Calculi and their Applications (BISCA)".

References

[1] L. Aceto, Deriving complete inference systems for a class of GSOS languages generating regular behaviors, in: Proceeding of the 5th International Conference on Concurrency Theory, CONCUR'94, in: Lecture Notes in Computer Science, vol. 836, Springer, 1994, pp. 449–464.

[2] L. Aceto, B. Bloom, F. Vaandrager, Turning SOS rules into equations, Inform. and Comput. 111 (1994) 1–52.

[3] L. Aceto, W.J. Fokkink, C. Verhoef, Structural operational semantics, in: J.A. Bergstra, A. Ponse, S.A. Smolka (Eds.), Handbook of Process Algebra, Elsevier, 2001, pp. 197–292.

[4] O. Andreai, G. Ciobanu, D. Lucanu, Structural operational semantics of P Systems, in: Proceedings of the 6th Workshop on Membrane Computing, WMC 2005, in Lecture Notes in Computer Science, vol. 3850, Springer, 2006, pp. 31–28.

[5] O. Andreai, G. Ciobanu, D. Lucanu, A rewriting logic framework for operational semantics of membrane systems, Theoret. Comp. Sci. 373 (2007) 163–181.

[6] J.A. Bergstra, J.W. Klop, Process algebra for synchronous communication, Inform. and Comput. 60 (1984) 109–137.

[7] G. Berry, A. Benveniste (Eds), Another look at real time systems, IEEE Proc. 79 (1991) 1268–1336.

[8] N. Busi, Using well–structured transition systems to decide divergence for catalytic P systems, Theoret. Comput. Sci. 372 (2007) 125–135.

[9] N. Busi, Causality in membrane systems, in: Proceedings of the 8th Workshop on Membrane Computing, WMC 2007, in: Lecture Notes in Computer Science, vol. 4860, Springer, 2007, pp. 160–171.

[10] R. de Simone, High level synchronization devices in Meije-SCCS, Theoret. Comput. Sci. 37 (1985) 245–267.

[11] R. Freund, S. Verlan, A formal framework for static (tissue) P Systems, in: Proceeding of the 8th Workshop on Membrane Computing, WMC 2007, in: Lecture Notes in Computer Science, vol. 4860, Springer, 2007, pp. 271–284.

[12] R. Keller, Formal verification of parallel programs, CACM 19 (1976) 371–384.

[13] G. Lüttgen, M. von der Beeck, R. Cleaveland, Statecharts via process algebra, in: Proceedings of the 10th International Conference on Concurrency Theory, CONCUR'99, in: Lecture Notes in Computer Science 1664, Springer, 1999, pp. 399–414.

[14] A. Maggiolo-Schettini, A. Peron, S Tini, A comparison of statecharts step semantics, Theoret. Comput. Sci. 290 (2003) 465–498.

[15] R. Milner, A complete inference system for a class of regular behaviors, J. Comput. System Sci. 28 (1984) 439–466.

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

[17] G. Păun, G. Rozenberg, A guide to membrane computing, Theoret. Comput. Sci. 287 (2002) 73–100.

[18] G. Plotkin, A structural spproach to operational semantics, Tech. Rep. DAIMI FN-19, University of Aarhus, 1981.

[19] G. Plotkin, A structural approach to operational semantics, J. Log. Algebr. Program. 60–61 (2004) 17–139.

[20] S. Tini, An axiomatic semantics for Esterel, Theoret. Comput. Sci. 269 (2001) 231–282.

[21] S. Tini, An axiomatic semantics for the synchronous language Gentzen, J. Comput. Syst. Sci. 66 (2003) 316–348.

[22] A.C. Uselton and S.A. Smolka, A compositional semantics for Statecharts using labeled transition systems, in: Proceeding of the 5th International Conference on Concurrency Theory, CONCUR'94, in: Lecture Notes in Computer Science, vol. 836, Springer, 1994, pp. 2–17.

[23] F. Vaandrager, On the relationship between process algebra and input/output automata, in: Proceedings of the 6th Annual IEEE Symposium on Logic in Computer Science, LICS'91, IEEE Press, 1991, pp. 387–398.

[24] R.J. van Glabbeek, Full abstraction in structural operational semantics, in: Proceedings of the 3rd International Conference on Algebraic Methodology and Software Technology, AMAST 1993, in: Workshops on Computing, Springer, 1993, pp. 77–84.