



Strategy-Based Rewrite Semantics for Membrane Systems Preserves Maximal Concurrency of Evolution Rule Actions

Dorel Lucanu^{1,2}

Faculty of Computer Science, Alexandru Ioan Cuza University, Iași, Romania

Abstract

We use a modal logic in order to show that the strategy-based rewrite semantics for membrane systems fully preserves the maximal concurrency of evolution rules actions, whereas the maximal concurrency of communication actions and structural actions is partially preserved. Consequently, the strategy-based rewrite semantics describes more faithfully the behavior of the membrane systems than the rewrite logic-based semantics. It is known that the rewrite logic-based semantics implements the maximal concurrency of the evolution rules in membrane systems only by interleaving concurrency. The concurrency degrees of the communication and structural actions are the same for the two rewrite-based semantics.

Keywords: Rewrite Strategies, Strategy Controller, Membrane System, Modal logic, True Concurrency, Rewrite Logic.

1 Introduction

Membrane computing [18] deals with distributed and parallel computing models inspired from the structure and the functioning of living cells, as well as from the way the cells are organized. Such a model processes multisets of symbol-objects in a localized manner. The locality of processing refers to the fact that the evolution rules and evolving objects are encapsulated into compartments delimited by membranes. An essential role is also played by the communication among compartments and, possibly, with the environment.

There are several approaches [3,1] which describe a rewrite semantics based on rewriting logic (RWL) [15,16]. Even if the use of RWL framework seems to be a natural choice for specifying and analyzing membrane systems, the locality of evolution rules and the higher degree of the concurrency given by the maximal parallel

¹ This work is partially supported by the PN II grant ID 393/2007 and CNCSIS 1162/2008.

² Email: dlucanu@info.uaic.ro

rewriting (used in defining the behavior of these systems) is quite challenging. An alternative approach based on rewrite strategies and strategy controllers is given in [4]. The main idea is to separate the implementation of the control mechanisms of regions from the effective application of the evolution rules.

In [13] we show that RWL-based semantics can describe the maximal parallel rewriting of the membrane systems only by interleaving semantics. In this paper we show that the strategy-based rewrite semantics defined in [4] preserves the maximal concurrency expressed by the maximal parallel application of the evolution rules. The concurrency degree of the communications and structural actions is the same in the RWL-based semantics and strategy-based rewrite semantics. Since the two formalisms, membrane systems and strategy-based rewriting logic, are quite different, we use a simple modal logic as a common language for comparing the concurrency degrees of the two formalisms.

The paper is structured as follows. Section 2 briefly presents the membrane systems and rewriting logic. In Section 3 a Hennessy-Milner-like modal logic for membrane systems is introduced. Section 4 briefly recalls from [4] the strategy-based rewrite semantics for membrane systems. It further includes an algorithm computing strategies for communications and dissolvings with the highest concurrency degree. Section 5 includes the main results of the paper. The concurrency degree of an evolution step of a membrane system is compared with that of its implementation as a strategic rewrite using the modal logic introduced in Section 3. The paper ends with some concluding remarks.

Acknowledgments: The author would like to thank the anonymous referees for their useful remarks and comments and to C. Croitoru for the helpful discussions regarding edge coloring problem for bipartite graphs.

2 Preliminaries

2.1 Membrane systems

In this paper we consider a particular case of membrane systems, namely that known as *transition P systems* [18]. Informally, a transition P system consists of: an alphabet of objects (usually a finite non-empty alphabet of symbols identifying the real objects), the membrane (region) structure (it can be represented in many ways, but the most used one is by a string of labeled matching parentheses), the multisets of objects present in each membrane of the system (represented in the most compact way by strings of symbol-objects), the sets of evolution rules associated with each region, as well as the indication about the way the output is defined (see, e.g., Figure 1). A membrane structure is a hierarchically arranged set of membranes, contained in a distinguished external membrane called the *skin* membrane. Several membranes can be placed inside a membrane; a membrane is called *elementary* if it has no other membrane inside it. Each membrane determines a compartment, also called region, the space delimited from above by it and from below by the membranes placed directly inside, if any exists. Clearly, the correspon-

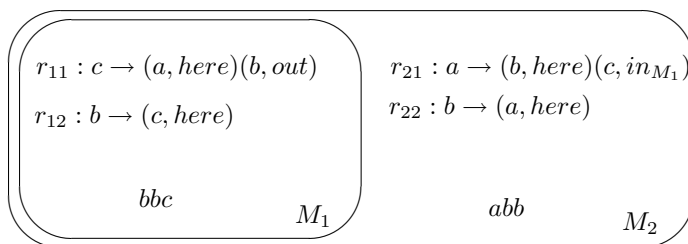


Fig. 1. A P system with two membranes

dence membrane-region is one-to-one, that is why we sometimes use interchangeably these terms. The hierarchical structure of membranes is a rooted tree symbolically represented as a string of labeled matching parentheses. The evolution rules have the form $r : u \rightarrow v$ or $r : u \rightarrow v\delta$, with u a non-empty multiset over O , v a multiset over $O \times Tar$, where $Tar = \{\text{here}, \text{out}\} \cup \{\text{in}_j \mid 1 \leq j \leq m\}$, and δ a special object called *dissolving action*. The elements of Tar are called *target indications* and have the following meaning: an object having associated the indication *here* remains in the same region, one having associated the indication in_j goes immediately into the directly lower membrane j , and *out* indicates that the object has to exit the membrane, thus becoming an element of the region surrounding it. Note that the set Tar is depending on the number of the membranes in the system. A pair $(a, tar) \in O \times Tar$ is often called *message*. A rule $r : u \rightarrow v$ can be cooperative (with u arbitrary), non-cooperative (with $u \in O \setminus C$), or catalytic (of the form $ca \rightarrow (c, \text{here})v$ or $ca \rightarrow (c, \text{here})v\delta$, with $a \in O \setminus C$, $c \in C$, and v a multiset over $(O \setminus C) \times Tar$); note that the catalysts never evolve and never change the region, they only help the other objects to evolve.

In this paper we associate a distinguished name M_j to each membrane j and the name M_j and the index j are used interchangeably.

Formally, a *transition P system* (of degree m) is a construct of the form $\Pi = (O, C, \mu, w_1, w_2, \dots, w_m, R_1, R_2, \dots, R_m, i_0)$, where:

- (i) O is the (finite and non-empty) alphabet of *objects*,
- (ii) $C \subseteq O$ is the set of *catalysts*,
- (iii) μ is a *membrane structure*, consisting of m membranes, labeled with $1, 2, \dots, m$; one says that the membrane structure, and hence the system, is of *degree m* ,
- (iv) w_1, w_2, \dots, w_m are multisets over O representing the multisets of objects present in the regions $1, 2, \dots, m$ of the membrane structure (*contents*),
- (v) R_1, R_2, \dots, R_m are finite sets of *evolution rules* associated with the regions $1, 2, \dots, m$ of the membrane structure,
- (vi) i_0 is either one of the labels $1, 2, \dots, m$, and then the respective region is the *output region* of the system, or it is 0, and then the result of a computation is collected in the environment of the system.

A *configuration* (μ, w_1, \dots, w_m) consists of the membrane structure μ of degree m and the multisets w_i of objects/messages from its compartments. During the evolution of the system, both the multisets of objects/messages and the membrane

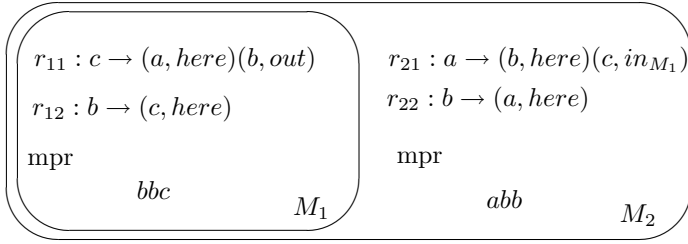


Fig. 2. A P system with control mechanisms

structure can change. For instance, the configuration of the system represented in Figure 1 is $([[[]_1]_2, bbc, aab)$. We represent the multisets as (commutative) strings.

The contents of the membranes evolve by means of evolution rules. In each time unit a transformation of a configuration of the system, called *evolution step*, takes place by applying the rules in each region, in a *non-deterministic and maximally parallel manner*. An evolution step in a given region (membrane) consists in 1) *finding a maximal applicable multiset of evolution rules*, removing from the region all objects specified in the left hand sides of the chosen rules (with the multiplicities as indicated by the rules and by the number of times each rule is used), producing the messages from the right hand sides of rules, and then 2) *distributing the objects from these messages* as indicated by their target component. If at least one of the rules introduces the dissolving action δ , then 3) *the membrane is dissolved*, and its content becomes part of the immediately upper membrane, provided that this membrane was not dissolved at the same time, a case where we stop in the first upper membrane which was not dissolved (at least the skin remains intact). The rules of the dissolved membranes are lost.

There are many extensions of transitional P systems, among which we mention here the use of priority relation over the evolution rules, the use of promoters and inhibitors, the non-deterministic choice of the *in* target. The reader is invited to see [18] for a detailed presentation. We associate to each membrane a *control mechanism* specifying its particular way to evolve (see, e.g., Figure 2). Different membranes of a system may have different control mechanisms. For the sake of presentation, in this paper we consider only the mechanism *mpr* given by the maximal parallel rewriting, where a maximal applicable multiset of evolution rules is used at each evolution step. The approach can be applied for other mechanisms, as well.

We briefly recall some notations regarding the formal operational semantics of membrane systems:

- *semantics of the control mechanisms*: $w \rightarrow_M w'$ whenever w' is obtained from w by applying the evolution rules of M according to the control mechanism of M . For instance, since the control of M_1 is maximal parallel rewriting (mpr), we have $bbc \rightarrow_{M_1} (c, \text{here})(c, \text{here})(a, \text{here})(b, \text{out})$, applying in parallel twice r_{12} and once r_{11} . Note that w is a multiset of objects and w' is a mixed multiset of objects and messages.
- *semantics of the evolution rules actions*: $(\mu, w_1, \dots, w_m) \rightarrow_{evrl} (\mu, w'_1, \dots, w'_m)$ whenever for each i , either w_i is irreducible and $w'_i = w_i$ or $w_i \rightarrow_{M_i} w'_i$. We assume that at least one w_i is reducible. For instance,

$([\Box_1]_2, bc, ab) \rightarrow_{evrl} ([\Box_1]_2, (c, here)(a, here)(b, out), (b, here)(c, in_1)(a, here))$.

- *semantics of the communication actions:* $(\mu, w_1, \dots, w_m) \rightarrow_{comm} (\mu, w'_1, \dots, w'_m)$ whenever there is at least one i such that w_i is a multiset over $O \times Tar$ (a message), and for all j , w'_j is obtained by distributing the objects from all w_i according to their targets. For instance, $([\Box_1]_2, (c, here)(a, here)(b, out), (b, here)(c, in_1)(a, here)) \rightarrow_{comm} ([\Box_1]_2, cac, bba)$.
- *semantics of the dissolving actions:* $(\mu, w_1, \dots, w_m) \rightarrow_{diss} (\mu', w'_1, \dots, w'_n)$ whenever there is at least one i such that $\delta \in w_i$, and μ' and $(w'_j)_j$ are obtained by dissolving all membranes i with $\delta \in w_i$. For instance, $([\Box_{M_1}]_{M_2}, ab\delta, ccd) \rightarrow_{diss} ([\Box_{M_2}], ccdab)$.

We often use \bar{w} for w_1, \dots, w_m , \bar{w}' for w'_1, \dots, w'_m and so on. An *evolution step* of a system is either of the form $(\mu, \bar{w}) \rightarrow_{evrl} (\mu, \bar{w}')$, where \bar{w}' has no messages or δ (no communications and no dissolvings, e.g., only evolution rules removing objects are applicable), or $(\mu, \bar{w}) \rightarrow_{evrl} (\mu, \bar{w}') \rightarrow_{comm} (\mu, \bar{w}'')$, where \bar{w}'' has no δ (no dissolvings), or $(\mu, \bar{w}) \rightarrow_{evrl} (\mu, \bar{w}') \rightarrow_{diss} (\mu, \bar{w}'')$, where \bar{w}'' has no messages (no communications, e.g., only rules of the form $u \rightarrow \delta$ are applicable), or $(\mu, \bar{w}) \rightarrow_{evrl} (\mu, \bar{w}') \rightarrow_{comm} (\mu, \bar{w}'') \rightarrow_{diss} (\mu, \bar{w}''')$.

2.2 Rewriting Logic (RWL)

We assume that the reader is familiar with the basic definitions and notations for many-sorted equational logic [9], term rewriting [5,6], membership equational logic (MEL) [17,7], and rewriting logic [8,15,16].

Here we consider only (*unconditional*) MEL-based rewrite theories $\mathcal{R} = (\Sigma, E, R)$, where

- (Σ, E) is a MEL theory consisting of a MEL signature Σ and a set E of MEL axioms (membership axioms and equations), and
- R is a set of (universally quantified) labeled (unconditional) rewrite rules having the form $(\forall X)r : u \rightarrow v$, with $u, v \in \mathbb{T}_\Sigma(X)_s$ ($=$ the set of terms of sort s and with variables in X) for some sort s and $Var(v) \subseteq Var(u) \subseteq X$.

The rewriting logic of a MEL-based rewrite theory \mathcal{R} consists of

- sentences given by *rewrite sequents*, which are pairs of the form $(\forall X)t \rightarrow t'$, with $t, t' \in \mathbb{T}_\Sigma(X)_s$ for some sort s , and
- an *entailment relation* $\mathcal{R} \vdash (\forall X)t \rightarrow t'$ defined by a set of inference rules (see, e.g., [8,15] for details).

A *one-step concurrent rewrite* is a sequent $(\forall X)t \rightarrow t'$ which can be obtained by a derivation (deduction) which does not use the sequential composition deduction rule. In other words, t' can be obtained from t by applying in parallel a multiset of rewrite rules.

We give as examples the rewrite theories describing the control-free membranes (no restrictions regarding the application of the evolution rules are considered).

The Rewrite Theory Associated to an Elementary Membrane. The static descrip-

tion of the membranes is represented by the MEL theory (Σ_m, E_m) , where Σ_m includes the sorts *Object*, *Soup*, *Tar*, and *HotSoup* with $\text{Object} < \text{Soup} < \text{HotSoup}$, $(w, \text{tar}) : \text{HotSoup}$ if $w : \text{Soup}$ and $\text{tar} : \text{Tar}$, $\varepsilon : \rightarrow \text{HotSoup}$, the concatenation $-- : \text{HotSoup } \text{HotSoup} \rightarrow \text{HotSoup}$. The sort *Tar* includes the constants *here* and *out*. The operation in_- will be defined later. E_m includes axioms expressing the associativity and commutativity of $--$ with ε the identity element, and an axiom which allows to merge two messages with the same target into a single message: $(w_1, \text{tar})(w_2, \text{tar}) = (w_1w_2, \text{tar})$. The sort *Soup* is for the multisets of objects and the sort *HotSoup* for the mixed multisets of objects and messages. The idea is as follows: a soup is heated (transformed into a hot soup) applying the evolution rules, then a hot soup is cooled (transformed back into a soup) distributing the objects from messages according to their targets. The complete description of a control-free membrane M is represented by the MEL rewrite theory $\mathcal{M} = (\Sigma_m \cup O, E_m, R)$, where O the set of object constants, and R includes the rewrite rules corresponding to the evolution rules. For the case of M_1 in Figure 1, O includes the constants a, b, c and R_1 includes the rules r_{11} and r_{12} . Using the inference rules for RWL, we may deduce, e.g., that the one-step evolution of M_1 $\text{bbc} \rightarrow_{M_1} (c, \text{here})(c, \text{here})(a, \text{here})(b, \text{out})$ can be described by an one-step concurrent rewrite:

$$\frac{(\forall \emptyset)b \rightarrow (c, \text{here}) \quad (\forall \emptyset)b \rightarrow (c, \text{here})}{\frac{bb \rightarrow (c, \text{here})(c, \text{here}) = (cc, \text{here}) \quad (\forall \emptyset)c \rightarrow (a, \text{here})(b, \text{out})}{bbc \rightarrow (cc, \text{here})(a, \text{here})(b, \text{out}) = (cca, \text{here})(b, \text{out})}}$$

Note that the above rewrite theory describes M_1 as an independent elementary membrane. We show below that the description of the behavior of M_1 by the rewrite theory corresponding to the whole system is more complicated.

The Rewrite Theory Associated to a Membrane System. The static description of membrane systems is represented by the MEL theory (Σ_p, E_p) consisting of (Σ_m, E_m) together with:

- a sort *MembraneName* together with a constant $M : \text{MembraneName}$, for each membrane name M , and an operation $\text{in}_- : \text{MembraneName} \rightarrow \text{Tar}$,
- a sort *Membrane* for states of both simple and composite membranes,
- a sort *MembraneBag* for multisets of membranes, together with its constructors: the subsort relation $\text{Membrane} < \text{MembraneBag}$, the constant *NULL* denoting the empty multiset, and the union of multisets
 $-, _- : \text{MembraneBag } \text{MembraneBag} \rightarrow \text{MembraneBag} \text{ [assoc comm id: NULL]}$
- the constructors for *Membrane*: $\langle - | - \rangle : \text{MembraneName } \text{HotSoup} \rightarrow \text{Membrane}$ and $\langle - | - \{ - \} \rangle : \text{MembraneName } \text{HotSoup } \text{MembraneBag} \rightarrow \text{Membrane}$, together with the axiom $\langle M | w \{ \text{NULL} \} \rangle = \langle M | w \rangle$.

A control-free membrane system Π (neither control mechanisms nor the membership of evolution rules to membranes is assumed) is described by the rewrite theory $\mathcal{R}_\Pi = (\Sigma_p \cup O, E_p, R)$, where R includes the rewrite rules coming from all the

component membranes together with the cooperation (interaction) rules (if any):

$$\begin{aligned}
 in(M, M') &: \langle M \mid w_1(w_2, in_{M'}) \{ \langle M' \mid w' \{ X \} \rangle, Y \} \rangle \rightarrow \\
 &\quad \langle M \mid w_1 \{ \langle M' \mid w' w_2 \{ X \} \rangle, Y \} \rangle \\
 out(M', M) &: \langle M \mid w \{ \langle M' \mid w'_1(w'_2, out) \{ X \} \rangle, Y \} \rangle \rightarrow \\
 &\quad \langle M \mid ww'_2 \{ \langle M' \mid w'_1 \{ X \} \rangle, Y \} \rangle \\
 diss(M') &: \langle M \mid w \{ \langle M' \mid w' \delta \{ X \} \rangle, Y \} \rangle \rightarrow \langle M \mid ww' \{ X, Y \} \rangle
 \end{aligned}$$

The first rule describes the transmission of a message from a parent membrane M to a child membrane M' , the second one the transmission of a message from a child membrane M' to the parent membrane M , and the third one the dissolving of the membrane M' (this is triggered by the presence of the object δ in the current content of M').

The rewrite theory \mathcal{R}_Π does not include information about the locality of the rewrite rules. For instance, if Π is the system described in Figure 1, then logic defined by \mathcal{R}_Π allows to apply r_{12} for both the content of M_1 and the content of M_2 . There are different ways for describing the locality of the evolution rules w.r.t. their membership to regions: considering an operation $rules(M)$ - returning the rules of the membrane M [3,4], encoding the set of rules in the description of each membrane [1,13] and so on. In either of these cases the rewriting logic fails to describe an one-step evolution of a membrane by an one-step concurrent rewrite. For instance, if we encode the rules of M_1 by $r_{11} : \langle M_1 \mid cW \rangle \rightarrow \langle M_1 \mid (a, here), (b, out)W \rangle$ and $r_{12} : \langle M_1 \mid bW \rangle \rightarrow \langle M_1 \mid (c, here)W \rangle$, then these two rules cannot be concurrently applied because they overlap.

3 A Modal Logic for Membrane Systems

In this section we define a Hennessy-Milner-like logic [11] able to express the concurrency degree (and the behavior) of a membrane system. We have seen that an evolution step consists of up to three transitions, where each transition accomplishes a specific task. Therefore we distinguish three kinds of actions in this logic:

- (i) *rule actions* corresponding to evolution rules of a membrane. Such an action is denoted by the label of the involved rule. We assume that the rules have distinguished labels such that there is no ambiguity regarding the rule or the membrane the rule belongs to.
- (ii) *communication actions* which describe how two parent-child membranes communicate. Here we consider two kinds of communications: $in(M, M')$ - the membrane M sends a message to membrane M' (M' is a child of M); and $out(M', M)$ - the membrane M' sends a message to the surrounding membrane M .
- (iii) *dissolving actions* $diss(M')$, meaning that the membrane M' is dissolved and its contents is sent to surrounding membrane; the evolution rules of M' are lost. In general, we may consider a more general kind, namely *structural actions* meaning all actions aimed to modify the structure of the system.

We associate a modal language \mathcal{L}_Π to a membrane system Π as follows:

- (i) *true* is a formula in \mathcal{L}_Π ;
- (ii) if φ is a formula in \mathcal{L}_Π and L a multiset of rule actions, then $\langle L \rangle \varphi$ is a formula in \mathcal{L}_Π ;
- (iii) if φ is a formula in \mathcal{L}_Π and C a set of communication actions, then $\langle C \rangle \varphi$ is a formula in \mathcal{L}_Π ;
- (iv) if φ is a formula in \mathcal{L}_Π and D a set of dissolving actions, then $\langle D \rangle \varphi$ is a formula in \mathcal{L}_Π ;
- (v) if φ_1 and φ_2 are formulas in \mathcal{L}_Π , then so are $\neg \varphi_1$ and $\varphi_1 \wedge \varphi_2$.

Intuitively, if A is a multiset of actions of the same kind, then a configuration (μ, \bar{w}) satisfies the formula $\langle A \rangle \varphi$ iff there exists a transition $(\mu, \bar{w}) \rightarrow_{tr} (\mu', \bar{w}')$ such that (μ', \bar{w}') is obtained by applying in parallel the actions A and (μ', \bar{w}') satisfies φ , where $tr \in \{evrl, comm, diss\}$ corresponds to the kind of the actions A .

The other propositional connectors are added to \mathcal{L}_Π in the usual way; e.g., *false* is the notation for $\neg true$. The modal operator $[A]\varphi$ is defined as $\neg \langle A \rangle \neg \varphi$, where A denotes a set of actions of the same type.

In the following Π is a system with the membranes M_1, \dots, M_m , μ and μ' range over the structure of Π , w_i, w'_i range over the contents of the membrane M_i , $\bar{w} = (w_1, \dots, w_m)$, $\bar{w}' = (w'_1, \dots, w'_m)$, L ranges over the nonempty multisets of rewrite actions, C over the nonempty sets of communication actions, and D over the nonempty sets of dissolving actions. The semantics of the modal formulas is as follows:

- (i) $\Pi, (\mu, \bar{w}) \models true$ for each (μ, \bar{w}) .
- (ii) $\Pi, (\mu, \bar{w}) \models \varphi_1 \wedge \varphi_2$ if and only if $\Pi, (\mu, \bar{w}) \models \varphi_1$ and $\Pi, (\mu, \bar{w}) \models \varphi_2$.
- (iii) $\Pi, (\mu, \bar{w}) \models \neg \varphi$ iff $\Pi, (\mu, \bar{w}) \not\models \varphi$.
- (iv) $\Pi, (\mu, \bar{w}) \models \langle L \rangle \varphi$ iff there is a transition $(\mu, \bar{w}) \rightarrow_{evrl} (\mu, \bar{w}')$ such that
 - if $w_i \rightarrow_{M_i} w'_i$ using the multiset of rules L_i , $1 \leq i \leq n$, then $L = \cup_{i=1}^n L_i$, and
 - $\Pi, (\mu, w'_1, \dots, w'_m) \models \varphi$.

In other words, L is the multiset of all rules involved in the evolution of all membranes; the multiplicity of a rule in L is equal to the number of times the rule was used in the evolution of its membrane.
- (v) $\Pi, (\mu, \bar{w}) \models \langle C \rangle \varphi$ iff there is a transition $(\mu, \bar{w}) \rightarrow_{comm} (\mu, \bar{w}')$ such that
 - $in(M_i, M_j) \in C$ iff M_j is a child of M_i and there is $(u_i, in_{M_j}) \in w_i$ (there are objects in the current content w_i of M_i to be sent to the child M_j),
 - $out(M_j, M_i) \in C$ iff M_j is a child of M_i and there is $(u_j, out) \in w_j$ (there are objects in the current content w_j of M_j to be sent to the parent M_i).
- (vi) $\Pi, (\mu, \bar{w}) \models \langle D \rangle \varphi$ iff there is a transition $(\mu, \bar{w}) \rightarrow_{diss} (\mu, \bar{w}')$ such that
 - $diss(M_j) \in D$ iff $w_j = v_j \delta$.

We often write $M, w \models \varphi$ for $\Pi, ([\]_M, w) \models \varphi$, i.e., Π consists of only the elementary membrane M . It is easy to see that if $\Pi, (\mu, \bar{w}) \models \langle L \rangle \varphi$, then L can be written as a disjoint union $L_1 \uplus \dots \uplus L_m$ such that $(\forall i) L_i \neq \emptyset$ implies $M_i, w_i \models \langle L_i \rangle true$.

For instance, if Π_{12} is the membrane system represented in Figure 1, then

$$\begin{aligned}
M_1, bbc &\models \langle r_{11}r_{12}r_{12} \rangle true & [iv] \\
M_2, abb &\models \langle r_{21}r_{22}r_{22} \rangle true & [iv] \\
\Pi_{12}, ([2]_1)_2, bbc, abb &\models \langle r_{11}r_{12}r_{12}r_{21}r_{22}r_{22} \rangle true & [iv] \\
\Pi_{12}, ([2]_1)_2, (c, here) \dots (b, out), (c, here)(c, in_1) \dots (a, here) &\models \\
&\quad \langle in(M_2, M_1) out(M_1, M_2) \rangle true & [v] \\
\Pi_{12}, ([2]_1)_2, bbc_1, abb_2 &\models \langle r_{11}r_{12}r_{12}r_{21}r_{22}r_{22} \rangle \langle in(M_2, M_1) out(M_1, M_2) \rangle true & [iv]
\end{aligned}$$

We have $\Pi, (\mu, \bar{w}) \models [A]\varphi$ if for all transitions $(\mu, \bar{w}) \rightarrow_{tr} (\mu', \bar{w}')$ such that (μ', \bar{w}') is obtained by applying in parallel the actions A and $tr \in \{evrl, comm, diss\}$ corresponds to the kind of the actions A , $\Pi, (\mu', \bar{w}') \models \varphi$. Since all evolution rules of M_1 are non-cooperative, we have $M_1, bbc \models \langle r_{11}r_{12}r_{12} \rangle true$ iff $M_1, bbc \models [r_{11}r_{12}r_{12}]true$, i.e., just one transition is possible from bbc . If, e.g., we add to M_1 the cooperative evolution rule $r_{13} : bc \rightarrow (b, here)(c, out)$, then we also have $M_1, bbc \models \langle r_{12}r_{13} \rangle true$. The all transitions possible from bbc now are characterized by $M_1, bbc \models \langle r_{11}r_{12}r_{12} \rangle true \wedge \langle r_{12}r_{13} \rangle true$ and $M_1, bbc \models [L]false$ if $L \neq r_{11}r_{12}r_{12}$ and $L \neq r_{12}r_{13}$. By $M_1, bbc \models [L]false$ we express that there is no any transition such that $bbc \rightarrow_{M_1} w'$ applying the evolution rules L' .

The definition of \mathcal{L}_Π allows legal formulas as $\langle L \rangle \langle D \rangle \langle C \rangle true$, which are non-satisfiable, i.e., there is no membrane systems satisfying such a formula. The definition of \mathcal{L}_Π can be strengthened in order to remove such constructions, but we preferred to let it as simple as possible.

Using the modal language \mathcal{L}_Π with the satisfaction relation previously defined, we are able to express the behavior of the system Π . An evolution step $(\mu, \bar{w}) \Rightarrow (\mu', \bar{w}')$ of a membrane system Π is described as follows:

- (i) no dissolvings, no communications: $((\mu, \bar{w}) \Rightarrow (\mu', \bar{w}'))$ iff $(\mu, \bar{w}) \rightarrow_{evrl} (\mu, \bar{w}')$; in that case there is a nonempty multiset L of rewrite actions such that $\Pi, (\mu, \bar{w}) \models \langle L \rangle true$ and for each nonempty set C of communications and nonempty set D of dissolvings, $\Pi, (\mu, \bar{w}') \models [C]false$ and $\Pi, (\mu, \bar{w}') \models [D]false$;
- (ii) no dissolvings: $((\mu, \bar{w}) \Rightarrow (\mu', \bar{w}'))$ iff $(\mu, \bar{w}) \rightarrow_{evrl} (\mu, \bar{w}') \rightarrow_{comm} (\mu, \bar{w}'')$; in that case there are a nonempty multiset L of rule actions and a nonempty set C of communications such that $\Pi, (\mu, \bar{w}) \models \langle L \rangle \langle C \rangle true$ and for each nonempty set D of dissolvings, $\Pi, (\mu, \bar{w}') \models [D]false$;
- (iii) no communications: $((\mu, \bar{w}) \Rightarrow (\mu', \bar{w}'))$ iff $(\mu, \bar{w}) \rightarrow_{evrl} (\mu, \bar{w}') \rightarrow_{comm} (\mu', \bar{w}'')$; in that case there are a multiset L of rule actions and a nonempty set D of dissolvings such that $\Pi, (\mu, \bar{w}) \models \langle L \rangle \langle D \rangle true$ and for each nonempty set C of communications, $\Pi, (\mu, \bar{w}') \models [C]false$;
- (iv) otherwise $((\mu, \bar{w}) \Rightarrow (\mu', \bar{w}'))$ iff $(\mu, \bar{w}) \rightarrow_{evrl} (\mu, \bar{w}') \rightarrow_{comm} (\mu, \bar{w}'') \rightarrow_{diss} (\mu', \bar{w}''')$; in that case there are a nonempty multiset L of rule actions, a nonempty set C of communications and a non-empty set D of dissolvings such that $\Pi, (\mu, \bar{w}) \models \langle L \rangle \langle C \rangle \langle D \rangle true$.

The following result exhibits that the behavior of the membrane systems with the mpr control mechanism exposes a maximal concurrency degree.

Proposition 3.1 *Let L be a nonempty multiset of rule actions, C a nonempty set*

of communication actions, and D a nonempty set of dissolving actions.

- (i) If $\Pi, (\mu, \bar{w}) \models \langle L \rangle \text{true}$, then $(\forall L' \neq \emptyset) \Pi, (\mu, \bar{w}) \models \langle L \rangle [L'] \text{false}$.
- (ii) If $\Pi, (\mu, \bar{w}) \models \langle C \rangle \text{true}$, then $(\forall C' \neq \emptyset) \Pi, (\mu, \bar{w}) \models \langle C \rangle [C'] \text{false}$.
- (iii) If $\Pi, (\mu, \bar{w}) \models \langle D \rangle \text{true}$, then $(\forall D' \neq \emptyset) \Pi, (\mu, \bar{w}) \models \langle D \rangle [D'] \text{false}$.

The proof of Proposition 3.1 follows direct from the definitions of \rightarrow_{evrl} , \rightarrow_{comm} , and \rightarrow_{diss} , respectively. For the case of the rule actions, does also mater the fact that the evolution rules consume objects and produce messages, so a new rule can be applied only after the objects are distributed according to their targets from messages, i.e., after the occurrences of the communication actions.

Since the implementation of this semantics is difficult in practice, we may derive from it other equivalent semantics but with different concurrency degrees:

- (i) *true concurrency semantics* \models_{tc} :
 - (a) if $\Pi, (\mu, \bar{w}) \models \langle A \rangle \varphi$, then $\Pi, (\mu, \bar{w}) \models_{tc} \langle A \rangle \varphi$;
 - (b) if $\Pi, (\mu, \bar{w}) \models_{tc} \langle A \rangle \varphi$ and $A_1 \uplus \dots \uplus A_n$ is a partition of A , then $\Pi, (\mu, \bar{w}) \models_{tc} \langle A_1 \rangle \dots \langle A_n \rangle \varphi$;
- (ii) *interleaving semantics* \models_{int} :
 - (a) if $\Pi, (\mu, \bar{w}) \models \langle A \rangle \varphi$ and $A = a_1 \dots a_n$, then $\Pi, (\mu, \bar{w}) \models_{int} \langle a_1 \rangle \dots \langle a_n \rangle \varphi$ (recall that $a_1 \dots a_n$ is a (multi)set, so the actions a_i can be written in any order),

where A ranges over multiset of rule actions, sets of communication actions, and sets of dissolving actions. Obviously, \models_{tc} and \models_{int} requires appropriate definitions for \rightarrow_{evrl} , \rightarrow_{comm} , and \rightarrow_{diss} , respectively. The above modal formula-based definitions for concurrency degrees are inspired from [12].

However, the modal language was designated as minimal with respect to the concurrency of the membrane systems. Besides the concurrency degree, there is other information which can be of interest regarding the current state: the structure of the system (relationships between regions), explicit description of the membership of a content to its own region, relationships between the objects of a content and so on.

4 Strategy-based Rewrite Logic for Membrane Systems

In this section we recall from [4] the definition for the strategy-based rewrite logic for membrane systems. We further present here an algorithm for maximal parallel communication step. The correctness of this algorithm is based on König Theorem for the edge coloring of bipartite graphs. In [4] only an interleaving semantics is considered for communications and dissolvings.

Strategy-based rewrite logic for membrane systems was defined in [4] and specifies a membrane system Π by a triple $\mathcal{SR}_\Pi = (\mathcal{R}_\Pi, \text{STRAT}_\Pi, \text{SCTRL}_\Pi)$, where $\mathcal{R}_\Pi = (\Sigma, E, R)$ is a rewrite theory that specifies the control-free system Π and defined as in Section 2.2, STRAT_Π specifies a strategy language for Π , and SCTRL_Π defines the strategy controllers for Π . A strategy controller is intended to equationally define the control mechanisms of Π . The rewrite strategies are used to guide

the rewriting according to the operational semantics of Π . The semantics of \mathcal{SR}_Π is given by a MEL theory $\mathcal{Proof}(\Pi)$, which includes the semantics of the strategies and the strategy controllers. We briefly describe here the last three theories.

The Equational Theory STRAT_Π . We consider a minimal strategy language able to express the computations of a membrane system.

STRAT_Π defines the syntax for strategies and consisting of:

- a sort *RuleLabel* for representing rules, together with a membership axiom $r : \text{RuleLabel}$, for each rule $r : u \rightarrow v$ in R ,
- a sort *Strategy* for strategies, and the subsort relation $\text{RuleLabel} < \text{Strategy}$,
- the strategy constructors for identity, failure, non-deterministic choice, and sequential composition respectively:

$$\begin{aligned} id \quad fail &: \longrightarrow \text{Strategy} \\ _+ \quad _- &: \text{Strategy} \text{ Strategy} \longrightarrow \text{Strategy} \text{ [assoc comm]} \\ _; &: \text{Strategy} \text{ Strategy} \longrightarrow \text{Strategy} \text{ [assoc id : id]} \end{aligned}$$

- a congruence strategy operator for each of the constructors of *Soup*, *Membrane* and *MembraneBag*:

$$\begin{aligned} _- &: \text{Strategy} \text{ Strategy} \longrightarrow \text{Strategy} \text{ [assoc comm]} \\ \langle _- \rangle &: \text{MembraneName} \text{ Strategy} \longrightarrow \text{Strategy} \\ \langle _- \{ _- \} \rangle &: \text{MembraneName} \text{ Strategy} \text{ Strategy} \longrightarrow \text{Strategy} \\ \rightarrow _- &: \text{Strategy} \text{ Strategy} \longrightarrow \text{Strategy} \text{ [assoc comm]} \end{aligned}$$

The strategy language defined by the above theory was designed having in mind mainly the control of the evolution rules. This language can be enriched with new constructs needed for defining other control mechanisms over evolution rules [2] or to add certain control over the interaction rules. In Section 5 we sketch out an extension for the case of cooperation rules.

The Equational Theory SCTRL_Π . SCTRL_Π is the MEL theory consisting of:

- a sort *StrategyController* - for *strategy controllers*, together with the constants *mpr* ... - corresponding to the control mechanisms, *evrl* - corresponding to evolution rules actions, *comm* - corresponding to communication actions, and *diss* - corresponding to dissolving actions,
- an operation $\text{getCtrl} : \text{MembraneName} \longrightarrow \text{StrategyController}$ which returns the constant corresponding to the control mechanism of the membrane.

The equational theory $\mathcal{Proof}(\Pi)$. The proof-theoretical semantics of the specification $(\mathcal{R}_\Pi, \text{STRAT}_\Pi, \text{SCTRL}_\Pi)$ is given by a MEL-theory $\mathcal{Proof}(\Pi)$ which includes the semantics for strategies and the semantics for strategy controllers.

The semantics of strategies can be defined in different ways. In this paper, in order to keep the presentation as simple as possible, we consider the set-theoretical semantics [14] defined by the following additional operations:

$$\llbracket _ @ _ \rrbracket : \text{Strategy State} \longrightarrow \text{Set}\{\text{States}\}$$

$$\llbracket _ @ _ \rrbracket : \text{Strategy Set}\{\text{States}\} \longrightarrow \text{Set}\{\text{States}\}$$

together with the following equations:

$$\llbracket id @ t \rrbracket = \{t\}$$

$$\llbracket fail @ t \rrbracket = \emptyset$$

$$\llbracket r @ t \rrbracket = \{t' \mid t \text{ rewritten directly modulo } E_p \text{ to } t' \text{ using the rule } r \text{ at top}\}$$

$$\llbracket s_1 + s_2 @ t \rrbracket = \llbracket s_1 @ t \rrbracket \cup \llbracket s_2 @ t \rrbracket$$

$$\llbracket s_1 ; s_2 @ t \rrbracket = \llbracket s_2 @ \llbracket s_1 @ t \rrbracket \rrbracket$$

$$\llbracket s_1 s_2 @ w_1 w_2 \rrbracket = \{w'_1 w'_2 \mid w'_i \in \llbracket s_i @ w_i \rrbracket, i = 1, 2\}$$

$$\llbracket \langle M \mid s \rangle @ \langle M \mid w \rangle \rrbracket = \{\langle M \mid w' \rangle \mid w' \in \llbracket s @ w \rrbracket\}$$

$$\llbracket s_1, s_2 @ t_1, t_2 \rrbracket = \{t'_1, t'_2 \mid t'_i \in \llbracket s_i @ t_i \rrbracket, i = 1, 2\}$$

$$\llbracket \langle M \mid s_1 \{s_2\} \rangle @ \langle M \mid w \{t\} \rangle \rrbracket = \{\langle M \mid w' \{t'\} \rangle \mid w' \in \llbracket s_1 @ w \rrbracket, t' \in \llbracket s_2 @ t \rrbracket\}$$

$$\llbracket s @ \emptyset \rrbracket = \emptyset$$

$$\llbracket s @ (T \cup \{t\}) \rrbracket = \llbracket s @ T \rrbracket \cup \llbracket s @ t \rrbracket$$

where *State* is a supersort of *HotSoup*, *Membrane* and *MembraneBag*, w_i, w'_i are variables of sort *Soup*, t, t', t_i, t'_i are variables of sort *State*, T a variable of sort *Set*{*States*}, and s, s_i are variables of sort *Strategy*.

However, we assume that the operational semantics for strategies is able to implement a strategy as a concurrent (parallel) rewriting whenever it is possible (e.g., see Corollary 4.5).

Definition 4.1 We say that two strategy terms s_1 and s_2 are equivalent in *Proof*(Π), written $s_1 \equiv s_2$, if and only if *Proof*(Π) $\vdash \llbracket s_1 @ t \rrbracket = \llbracket s_2 @ t \rrbracket$ for all state terms t .

Proposition 4.2 [4] *The following equivalences are true in Proof(Π):*

$$s + s \equiv s$$

$$s ; fail \equiv fail ; s \equiv fail$$

$$\langle M \mid s + s' \rangle \equiv \langle M \mid s \rangle + \langle M \mid s' \rangle$$

$$\langle M \mid s_1 + s'_1 \{s_2\} \rangle \equiv \langle M \mid s_1 \{s_2\} \rangle + \langle M \mid s'_1 \{s_2\} \rangle$$

$$\langle M \mid s_1 \{s_2 + s'_2\} \rangle \equiv \langle M \mid s_1 \{s_2\} \rangle + \langle M \mid s_1 \{s'_2\} \rangle$$

$$id \ id \equiv id$$

$$s + fail \equiv s$$

$$s \ fail \equiv fail \ s \equiv fail$$

$$s_1 (s_2 + s'_2) \equiv s_1 \ s_2 + s_1 \ s'_2$$

$$s_1, (s_2 + s'_2) \equiv s_1, s_2 + s_1, s'_2$$

The semantics of the strategy controllers is given by means of two operations:

$$getStrat : \text{StrategyController State} \longrightarrow \text{Strategy.}$$

$$getStrat : \text{MembraneName StrategyController State} \longrightarrow \text{Strategy}$$

If for a given state t there are possible more than one evolution steps, then $getStrat(ctrl, t)$ returns a sum of strategies expressing this nondeterminism. Therefore the strategy returned by $getStrat$ is unique up to \equiv .

The semantics of the strategy controller *evrl* is given by

$$\begin{aligned}
\text{getStrat}(\text{evrl}, \langle M \mid w \rangle) &= \langle M \mid \text{getStrat}(M, \text{getCtrl}(M), w) \rangle \\
\text{getStrat}(\text{evrl}, \langle M \mid w \{ t_1, \dots, t_n \} \rangle) &= \\
&\langle M \mid \text{getStrat}(M, \text{getCtrl}(M), w) \{ \text{getStrat}(\text{evrl}, t_1), \dots, \text{getStrat}(\text{evrl}, t_n) \} \rangle
\end{aligned}$$

The first equation defines *getStrat* for the elementary membranes and the second one for the case when the membrane M includes other membranes.

The strategy controllers *comm* and *diss* used in the implementation presented in [4] have an interleaving semantics. In order to capture the maximal concurrency degree for the communication (resp. dissolving) rewrite rules, we have to add to the strategy language a new operator

$$\langle \dots \rangle : \text{Strategy} \dots \text{Strategy} \longrightarrow \text{Strategy}$$

with the intuitive semantics that $\llbracket \langle s_1, \dots, s_n \rangle @ t \rrbracket$ is the set of terms which can be obtained from t by applying in parallel the strategies s_1, \dots, s_n at non-overlapping positions in t .

Let t be a term encoding a configuration (μ, \bar{w}) and let k be the maximum degree of μ (viewed as a tree). The tree μ is a bipartite graph and therefore it can be edge colored with k colors (by König Theorem, see, e.g., [10]) and this is the minimum of colors which can be used for a edge coloring. For each color c_ℓ , we consider two strategies $s_\ell = \langle \dots r \dots \rangle$ and $\bar{s}_\ell = \langle \dots \bar{r} \dots \rangle$ such that:

- (i) if r is $\text{in}(M_i, M_j)$ then \bar{r} is $\text{out}(M_j, M_i)$ and vice-versa;
- (ii) r occurs in s_ℓ iff \bar{r} occurs in \bar{s}_ℓ ;
- (iii) $\text{in}(M_i, M_j)$ occurs in s_ℓ or in \bar{s}_ℓ iff there is an edge (i, j) colored with c_ℓ .

It is obvious that s_ℓ and \bar{s}_ℓ have the same number of rule labels occurrences and the two strategies send the messages between the same set of membrane pairs but in opposite directions: if one sends messages from M_i to M_j , the other one sends from M_j to M_i . The rules specified by s_ℓ , respectively \bar{s}_ℓ , can be applied concurrently, due to the edge coloring properties. Therefore a strategy implementing *comm* with a maximum concurrency degree is $s_1; \bar{s}_1; \dots; s_k; \bar{s}_k$.

Then an algorithm computing $\text{getStrat}(\text{comm}, t) = s_1; \bar{s}_1; \dots; s_k; \bar{s}_k$ is an equational description of the edge coloring algorithm applied on the particular case of trees and where the colors are expressed as strategies.

In order to understand better this algorithm, we consider a simple example. Let Π be a membrane system with the tree structure $\mu = [[[[1]_2[[3]_4]_5]$. An edge coloring for μ is given by two colors: $\{(2, 1), (5, 4)\}$ and $\{(5, 2), (4, 3)\}$. A configuration for Π is of the form $t = \langle M_5 \mid w_5 \{ \langle M_2 \mid w_2 \{ \langle M_1 \mid w_1 \rangle \} \rangle, \langle M_4 \mid w_4 \{ \langle M_3 \mid w_3 \rangle \} \} \rangle$. A possible strategy returned by $\text{getStrat}(\text{comm}, t)$ is $s_1; \bar{s}_1; s_2; \bar{s}_2$, where:

$$\begin{aligned}
s_1 &= \langle \text{in}(M_2, M_1) \text{in}(M_5, M_4) \rangle & \bar{s}_1 &= \langle \text{out}(M_1, M_2) \text{out}(M_4, M_5) \rangle \\
s_2 &= \langle \text{in}(M_5, M_2) \text{out}(M_3, M_4) \rangle & \bar{s}_2 &= \langle \text{out}(M_2, M_5) \text{in}(M_4, M_3) \rangle
\end{aligned}$$

The following result summarizes the above discussion:

Proposition 4.3 *Let Π be a membrane system, t a term encoding a configuration (μ, \bar{w}) , k the maximum concurrency degree of μ . If $\text{Proof}(\Pi) \vdash \text{getStrat}(\text{comm}, t) = s$, then $s = s_1; \dots; s_{2k}$ and for each i , if $t' \in \llbracket s_i @ t \rrbracket$, then t' can be obtained from t*

by a concurrent one-step rewriting.

The algorithm for $\text{getStrat}(\text{diss}, t)$ is similar.

Regarding the control mechanisms of an elementary membrane, in this paper we are interested in the strategy controller mpr . The semantics of mpr , $\text{getStrat}(M, \text{mpr}, w)$ with $w : \text{Soup}$, is given by the means of two auxiliary operations

$$\begin{aligned} \text{mpr} : \text{RuleSet} &\longrightarrow \text{StrategyController} \\ \text{mpr} : \text{RuleSet } \text{Soup } \text{RuleSet } \text{Strategy} &\longrightarrow \text{StrategyController} \end{aligned}$$

together with the following equations:

$$\begin{aligned} \text{getStrat}(M, \text{mpr}, w) &= \text{getStrat}(\text{mpr}(\text{getRules}(M)), w) \\ \text{getStrat}(\text{mpr}(RS), w) &= \text{getStrat}(\text{mpr}(RS, w, \text{none}, \text{fail}), w) \\ \text{getStrat}(M, \text{mpr}((RS, r), w, RS', S), w) &= \\ & (r \text{ getStrat}(M, \text{mpr}((RS, r), w', \text{none}, \text{id}))) + \text{getStrat}(\text{mpr}((RS, r), w, (RS', r), \text{fail}), w) \\ & \text{if } \text{lhs}(r)w' := w \wedge \text{notIn}(r, RS') \\ \text{getStrat}(M, \text{mpr}(RS, w, RS', S), w) &= S[\text{owise}]. \end{aligned}$$

The attribute $[\text{owise}]$ used in the last equation means that this equation is conditional, where the condition accumulates the cases when none of the similar equations can be applied.

The third argument of the quaternary operation mpr represents the set of rules already used for the construction of a strategy applicable to w ; such iterative application allows to equally consider each of the rules given in the first argument for computing a strategy. As soon as a rule is tested as applicable and is used in a concatenation for building a strategy, then the fourth argument of mpr becomes id to mark the success. When the second and the third equations can no longer be applied, the fourth one is applied: if last argument of mpr is fail , this means no rule was applicable, and the strategy computed is fail by Proposition 4.2; otherwise, the strategy id is concatenated to the multiset of rules already found as applicable.

An evolution step in strategy-based rewrite semantics is defined as follows:

- $t \Rightarrow_{\text{evrl}} t_1$ iff $\text{Proof}(\Pi) \vdash t_1 \in \llbracket \text{getStrat}(\text{evrl}, t) @ t \rrbracket$,
- $t_1 \Rightarrow_{\text{comm}} t_2$ iff $\text{Proof}(\Pi) \vdash t_2 \in \llbracket \text{getStrat}(\text{comm}, t_1) @ t_1 \rrbracket$, and
- $t_2 \Rightarrow_{\text{diss}} t_3$ iff $\text{Proof}(\Pi) \vdash t_3 \in \llbracket \text{getStrat}(\text{diss}, t_2) @ t \rrbracket$.

The following result is a direct consequence of the definition of $\text{getStrat}(M, \text{ctrl}, w)$ and of Proposition 4.2.

Proposition 4.4 [4] *Let M be a membrane. If w is a term of sort Soup and $\text{Proof}(\Pi) \vdash \text{getStrat}(M, \text{mpr}, w) = s$ with $s \neq \text{fail}$, then s is equivalent to a sum of strategy terms s_i of the form $r_{i_1} \dots r_{i_n} \text{id}$, where $r_{i_1} \dots r_{i_n}$ is a multiset of rule labels. Moreover, $\llbracket s_i @ w \rrbracket \neq \emptyset$.*

Corollary 4.5 *If $\text{Proof}(\Pi) \vdash \text{getStrat}(M, \text{mpr}, w) = s$ with $s \neq \text{fail}$ and $w' \in \llbracket s @ w \rrbracket$, then w' can be obtained from w by a concurrent one-step rewriting.*

The above corollary is essential for the main result proved in the next section.

5 Concurrency in Strategy-based Rewrite Semantics

This section includes the main results of the paper. We use the operational correspondence between a membrane system and its strategy-based rewrite semantics [4] to show how the modal formulas can be used for expressing the concurrency degree for computations in the strategy-based rewrite semantics. Then we compare this concurrency degree with that of the original membrane system.

Throughout of this section we consider a membrane system Π , its representation $\mathcal{SR}_\Pi = (\mathcal{R}_\Pi, \text{STRAT}_\Pi, \text{SCTRL}_\Pi)$ as strategy-based rewrite theory, and $\text{Proof}(\Pi)$ the theory giving semantics to \mathcal{SR}_Π .

The static relationship between Π and \mathcal{SR}_Π is given by a function ψ defined as follows:

- (i) if w is a multiset of objects, then $\psi(w)$ is the corresponding ground term of sort *Soup*, denoted also by w (recall that each object of Π is defined as a constant of sort *Soup* in (Σ_p, E_p));
- (ii) if M is an elementary membrane with content w , then $\psi(M) = \langle M \mid w \rangle$;
- (iii) if M is a compound membrane with content w and the children M_1, \dots, M_n , then $\psi(M) = \langle M \mid w \{ \psi(M_1), \dots, \psi(M_n) \} \rangle$;
- (iv) $\psi(\mu, \bar{w}) = \psi(M)$, where M is the skin of Π (the relationship between \bar{w} and $\psi(M)$ is implicitly given here by means of μ).

The operational semantics correspondence is given by

$$\begin{aligned} (\mu, \bar{w}) \rightarrow_{ctrl} (\mu', \bar{w}') \text{ iff } & \psi(\mu, \bar{w}) \Rightarrow_{ctrl} \psi(\mu', \bar{w}') \\ & \text{iff } \text{Proof}(\Pi) \vdash \psi(\mu', \bar{w}') \in \llbracket \text{getStrat}(ctrl, \psi(\mu, \bar{w})) @ \psi(\mu, \bar{w}) \rrbracket \end{aligned}$$

where $ctrl \in \{evrl, comm, diss\}$. See [4] for more details.

The parallelism of the strategy in the strategy-based semantics is supplied by the congruence operators corresponding to concatenation (union) of multisets of soups and membranes, respectively. In order to express this parallelism, we associate a modal formula $\psi'(s)$ to a strategy term s as follows:

- (i) $\psi'(id) = true$, $\psi'(fail) = false$;
- (ii) $\psi'(r) = \langle r \rangle true$ if r is the label of a evolution/communication/structural rewrite rule;
- (iii) $\psi'(s_1 + s_2) = \psi'(s_1) \wedge \psi'(s_2)$;
- (iv) $\psi'(s_1; s_2) = \langle A_1 \rangle \varphi_2$ if $\psi'(s_1) = \langle A_1 \rangle true$ and $\psi'(s_2) = \varphi_2$;
- (v) $\psi'(s_1 s_2) = \langle A_1 \uplus A_2 \rangle (\varphi_1 \wedge \varphi_2)$ if $\psi'(s_i) = \langle A_i \rangle \varphi_i$ for $i = 1, 2$;
- (vi) $\psi'(\langle M \mid s \rangle) = \psi'(s)$;
- (vii) $\psi'(\langle M \mid s_1 \{s_2\} \rangle) = \langle A_1 \uplus A_2 \rangle (\varphi_1 \wedge \varphi_2)$ if $\psi'(s_i) = \langle A_i \rangle \varphi_i$ for $i = 1, 2$;
- (viii) $\psi'(s_1, s_2) = \langle A_1 \uplus A_2 \rangle (\varphi_1 \wedge \varphi_2)$ if $\psi'(s_i) = \langle A_i \rangle \varphi_i$ for $i = 1, 2$.

We recall that the rule labels are unique and \mathcal{SR}_Π stores the membership of rules to membranes, so we can recover the information lost by putting all the rule actions from different nesting levels in the same multiset. The function ψ' is partial; for instance, $\psi'(s_1; s_2)$ is not defined for all s_1 . Moreover, in order to have defined formulas like $\psi'(s id)$ we assume that $\varphi \equiv \langle \emptyset \rangle \varphi$.

5.1 Concurrency of Evolution Rules Actions

We have all the elements to compare the concurrency degrees of a membrane system Π and its strategy-based rewrite specification $(\mathcal{R}_\Pi, \text{STRAT}_\Pi, \text{CTRL}_\Pi)$.

Lemma 5.1 *Let M be an elementary membrane and w its content. If $\text{Proof}(\Pi) \vdash \text{getStrat}(\text{evrl}, \langle M|w \rangle) = s$, then $\psi'(s)$ is well-defined and has the form $\langle L_1 \rangle \text{true} \wedge \dots \wedge \langle L_k \rangle \text{true}$.*

The proof of the above result follows directly from Proposition 4.4.

Lemma 5.2 *Let M be an elementary membrane and w its content. $M, w \models \langle L \rangle \text{true}$ if and only if $\text{Proof}(\Pi) \vdash \text{getStrat}(\text{evrl}, \langle M|w \rangle) = s$ and there is φ such that $\psi'(s) = \varphi \wedge \langle L \rangle \text{true}$.*

Proof. We assume that $L = \{r_1, \dots, r_n\}$. If $M, w \models \langle L \rangle \text{true}$, then $s = s' + r_1 \dots r_n \text{id}$ from the definition of getStrat . The converse implication follows by applying Proposition 4.4 and Lemma 5.1. \square

Now we are able to prove the first main result of this paper:

Theorem 5.3 *Let L be a nonempty multiset of evolution rules actions.*

- 1) *If $\Pi, (\mu, \bar{w}) \models \langle L \rangle \text{true}$ then there are the strategy terms s, s' such that $\psi'(s') = \langle L \rangle \text{true}$ and $\text{Proof}(\Pi) \vdash \text{getStrat}(\text{evrl}, \psi(\mu, \bar{w})) = s + s'$.*
- 2) *Conversely, if $\text{Proof}(\Pi) \vdash \text{getStrat}(\text{evrl}, \psi(\mu, \bar{w})) = s$ then $\psi'(s)$ is well-defined and $\Pi, (\mu, \bar{w}) \models \psi'(s)$.*

Proof. If $\text{Proof}(\Pi) \vdash \text{getStrat}(\text{evrl}, \psi(\mu, \bar{w})) = s$, then s is a sum of strategies by Proposition 4.2 and each member of the sum defines a transition. The conclusions of the theorem follows by the operational semantics correspondence and Lemma 5.2. \square

It is worth noting that if $\text{Proof}(\Pi) \vdash \text{getStrat}(\psi(\Pi, \bar{w})) = s$ and s_i is a sum member of s , then s_i can be implemented by a concurrent one-step \mathcal{R}_Π -rewriting (see Corollary 4.5). Therefore Theorem 5.3 says that *the maximal concurrency of the rewrite actions is preserved by the strategy-based rewrite semantics*.

5.2 Concurrency Of Communication Actions

Let us consider first a parent-child pair (M, M') of membranes such that there are messages in M to be sent to M' and, conversely, there are messages in M' to be sent to M . In other words, the communication between them is described by the modal formula $\langle \text{in}(M, M') \text{ out}(M', M) \rangle \text{true}$. The communication between M and M' can be implemented by interleaving the *in* and *out* rules: $s = \text{in}(M, M'); \text{out}(M', M)$ or $s' = \text{out}(M', M); \text{in}(M, M')$. We have $\psi'(s) = \langle \text{in}(M, M') \rangle \langle \text{out}(M', M) \rangle \text{true}$.

This can be extended to the general case. We have seen that the algorithm presented on page 13 computes strategies which apply the communication rules in an concurrent way and as much as it is possible. In fact, by Proposition 4.3, we have that the *maximum concurrency degree of the communication which can be*

described in the rewrite semantics is that given by rewriting logic. This is because the communication rewrite rules are global and not local as it is the case of evolution rules.

We conclude now the second main result, namely that *the true concurrency of the communication actions is partially preserved by the strategy-based rewrite semantics, i.e., it is described by a combination of true concurrency and interleaving concurrency*. This is formalized by the following result:

Theorem 5.4 *Let Π be a membrane system and (μ, \bar{w}) a configuration of Π such that $\Pi, (\mu, \bar{w}) \models \langle C \rangle \text{true}$, where C is a nonempty set of communication actions. Let s be the strategy such that $\text{Proof}(\Pi) \vdash \text{getStrat}(\text{comm}, \psi(\mu, \bar{w})) = s$. Then there is a partition $C_1 \uplus \dots \uplus C_n$ of C such that $\psi'(s) = \langle C_1 \rangle \dots \langle C_n \rangle \text{true}$.*

Since $\Pi, (\mu, \bar{w}) \models \langle C \rangle \text{true}$, it follows that Π has at least two membranes. The proof of Theorem 5.4 is a direct consequence of the algorithm presented on page 13. It is worth noting that if C says that there are parent-child membranes which do not communicate or communicate in only one direction, *in* or *out*, then n given by Theorem 5.4 could be less than $2k$, where k is the maximum degree of μ . If we add to \mathcal{R}_Π the following rule, which simultaneously exchanges the messages between two membranes:

$$\begin{aligned} \text{in-out}(M, M') : \langle M \mid w_1(w_2, \text{in}_{M'}) \{ \langle M' \mid w'_1(w'_2, \text{out}) \{ X \} \}, Y \} \rangle \rightarrow \\ \langle M \mid w_1 w'_2 \{ \langle M' \mid w'_1 w_2 \{ X \} \}, Y \} \rangle \end{aligned}$$

and $\psi'(\text{in-out}(M, M')) = \langle \text{in}(M, M') \text{out}(M', M) \rangle$, then we get $n \leq k$.

5.3 Concurrency of Structural Actions

If the structural actions include only dissolvings, then we get a similar conclusion as the one for the communication actions. However, since the structural actions change the structure of the system, the maximal concurrency of structural actions cannot always be described by interleaving concurrency in the strategy-based rewrite semantics. For the case of dissolvings, because we consider only actions with one parameter, the label of the dissolving membrane, the interleaving is preserved. For instance, the double-dissolving in the configuration $([1[2[3]3]2]_1, w_1, w_2\delta, w_3\delta)$ could be described by one of the following two strategies:

$$s = \text{diss}(M_2); \text{diss}(M_3) \text{ or } s' = \text{diss}(M_3); \text{diss}(M_2).$$

Obviously $\psi'(s + s')$ expresses the interleaving of the two dissolvings. This does not remain true if the dissolving actions are of the form $\text{diss}(M, M')$.

Theorem 5.5 *Let Π be a membrane system and (μ, \bar{w}) a configuration of Π such that $\Pi, (\mu, \bar{w}) \models \langle D \rangle \text{true}$, where D is a nonempty set of dissolving actions. Let s be the strategy such that $\text{Proof}(\Pi) \vdash \text{getStrat}(\text{diss}, \psi(\mu, \bar{w})) = s$. Then there is a partition $D_1 \uplus \dots \uplus D_n$ of D such that $\psi'(s) = \langle D_1 \rangle \dots \langle D_n \rangle \text{true}$.*

6 Conclusion

In this paper we give a partial answer to the question if it is possible to define a rewrite semantics for the membrane systems. It was recently shown [13] that rewriting logic-based semantics cannot preserve the maximal concurrency of the rewrite actions. The main reason is the locality of the evolution rules w.r.t. their membership to the regions. The rewrite rules encoding the evolution rules belonging to a region must share this locality and hence they cannot be applied concurrently.

In this paper we show that the strategy-based rewrite semantics introduced in [4] preserves the maximal concurrency of the rewrite actions. In the strategy-based rewrite semantics the control mechanisms of the membranes are modeled by strategy controllers. A strategy controller analyzes the current state and computes a strategy term describing all possible transitions from the current state. The strategy term corresponding to a membrane can be computed in such a way it preserves the concurrency degree given by the control mechanism. The semantics of the strategy controllers is equationally defined and therefore it does not affect the behavior described by the strategy-based theory.

Regarding the concurrency of the cooperation (communication and structural) actions, the two rewrite semantics are equivalent. Since the rewrite rules governing these actions are global, the concurrency given by the rewriting logic is the maximum we can obtain.

In a recent paper, Șerbănuță et al. [20] investigate the same problem of the faithful implementation of the membranes but using the framework K [19]. Their result is based on the following two facts:

- 1) a special encoding of the P systems by tagging the rewrite rules and the objects with the path in the structure-tree from the root to the membrane M , and
- 2) the rewriting rules in K can be applied concurrently even when they overlap, assuming that they do not change the overlapped portion of the term (may overlap on “read only” parts).

In this paper we consider the particular case of transition P systems [18]. There is a large variety of P systems. We think that the strategy-based rewrite semantics can faithfully describe almost all mechanisms used for controlling the evolution rules. It remains to investigate what happens with the concurrency degree of the cooperation actions for different more general structures, e.g., tissue-like structures or neural-like structures.

References

- [1] Agrigoroaiei, O. and G. Ciobanu, *Rewriting Logic Specification of Membrane Systems with Promoters and Inhibitors*, in: *Proceedings of WRLA*, 2008, pp. 1–16, to appear in ENTCS.
- [2] Andrei, O., G. Ciobanu and D. Lucanu, *Expressing Control Mechanisms in P systems by Rewriting Strategies*, in: H. J. Hoogeboom, G. Paun, G. Rozenberg and A. Salomaa, editors, *Workshop on Membrane Computing*, Lecture Notes in Computer Science **4361** (2006), pp. 154–169.
- [3] Andrei, O., G. Ciobanu and D. Lucanu, *A rewriting logic framework for operational semantics of membrane systems*, Theoretical Computer Science **373** (2007), pp. 163 – 181.

- [4] Andrei, O. and D. Lucanu, *Strategy-Based Proof Calculus for Membrane Systems*, in: *Proceedings of WRLA*, 2008, pp. 17–34, to appear in ENTCS.
- [5] Baader, F. and T. Nipkow, “Term Rewriting and All That.” Cambridge University Press, 1998.
- [6] Bezem, M., J. Klop and R. de Vrijer, editors, “Term Rewriting Systems.” Cambridge Tracts in Theoretical Computer Science **I**, Cambridge University Press, 2003.
- [7] Bouhoula, A., J.-P. Jouannaud and J. Meseguer, *Specification and proof in membership equational logic*, Theoretical Computer Science **236** (2000), pp. 35–132.
- [8] Bruni, R. and J. Meseguer, *Semantic foundations for generalized rewrite theories*, Theoretical Computer Science **360** (2006), pp. 386–414.
- [9] Goguen, J. A. and J. Meseguer, *Order-Sorted Algebra I: Equational Deduction for Multiple Inheritance, Overloading, Exceptions and Partial Operations*, Theoretical Computer Science **105** (1992), pp. 217–273.
- [10] Gross, J. L. and J. Yellen, “Handbook of Graph Theory,” CRC Press, 2003.
- [11] Hennessy, M. and R. Milner, *Algebraic laws for nondeterminism and concurrency*, J. of ACM **32** (1985), pp. 137–161.
- [12] Lodaya, K., M. Mukund, R. Ramanujam and P. Thiagarajan, *Models and logics for true concurrency*, in: P. Thiagarajan, editor, *Some Models and Logics for Concurrency*, Advanced School on the Algebraic, Logical and Categorical Foundations of Concurrency, Gargnano del Garda, 1991 .
- [13] Lucanu, D., *Rewriting Logic-based Semantics of Membrane Systems and the Maximal Concurrency*, in: *Proceedings of Prague International Workshop on Membrane Computing* (2008), pp. 23–34.
- [14] Martí-Oliet, N., J. Meseguer and A. Verdejo, *A Rewriting Semantics for Maude Strategies*, in: *Proceedings of WRLA*, 2008, pp. 207–226, to appear in ENTCS.
- [15] Meseguer, J., *Conditional Rewriting Logic as a Unified Model of Concurrency*, Theoretical Computer Science **96** (1992), pp. 73–155.
- [16] Meseguer, J., *Rewriting Logic as a Semantic Framework for Concurrency: a Progress Report*, in: *Proceedings of the 7th International Conference on Concurrency Theory*, Lecture Notes in Computer Science **1119**, 1996, pp. 331–372.
- [17] Meseguer, J., *Membership algebra as a logical framework for equational specification*, in: Francesco Parisi-Presicce, editor, *WADT*, Lecture Notes in Computer Science **1376** (1997), pp. 18–61.
- [18] Paun, G., “Membrane Computing. An Introduction,” Springer, 2002.
- [19] Roşu, G., *K: A Rewriting-Based Framework for Computations – Preliminary version*, Technical Report Department of Computer Science UIUCDCS-R-2007-2926 and College of Engineering UILU-ENG-2007-1827, University of Illinois at Urbana-Champaign (2007).
- [20] Şerbănuţă, T. F., G. Ştefănescu and G. Roşu, *Defining and executing P-systems with structured data in K*, in: *WMC’08*, 2008, to appear.