Journal of Universal Computer Science, vol. 23, no. 7 (2017), 692-703
submitted: 31/3/17, accepted: 30/6/17, appeared: 28/7/17 © J.UCS

Reversibility in Parallel Rewriting Systems

Bogdan Aman
(Romanian Academy, Institute of Computer Science
Blvd. Carol I no.8, 700505 Iaşi, Romania
bogdan.aman@iit.academiaromana-is.ro)

Gabriel Ciobanu
("A.I.Cuza" University, Faculty of Computer Science
Blvd. Carol I no.11, 700506 Iaşi, Romania
Romanian Academy, Institute of Computer Science
Blvd. Carol I no.8, 700505 Iaşi, Romania
gabriel@info.uaic.ro)

Abstract: This paper represents a study of reversibility in parallel rewriting systems over multisets. It emphasizes the controlled reversibility for a particular case of parallel rewriting systems given by membrane systems, a formalism inspired by the cell activity. We define reversible membrane systems in which the scenarios based on regular expressions are able to control the direction (forward or backward) of the evolution. The backward computation is triggered by a special symbol ρ introduced into the system. Several results concerning the evolution of membrane systems and reversible membrane systems are provided, opening new research opportunities.

Key Words: controlled reversibility, membrane computing
Category: F.1.1, F.4.2, F.4.3

1 Introduction

Since many biological phenomena are naturally reversible, reversible computation has biological motivations. There exist reversible formalisms to model biological systems [Cardelli and Laneve 2011] as well as chemical reactions [Kuhn and Ulidowski 2016]. Reversible computation could be a suitable abstraction for a range of application domains (e.g., transactions and fault tolerant systems).

It is worth pointing out that there exist both uncontrolled and controlled reversibility. Uncontrolled reversibility indicates how to manage a system to reverse it to a previous state (without indicating when a backward evolution is required). This uncontrolled reversibility can help us to understand how reversibility works, without looking for some applications. However, in chemical and biological systems certain operations are reversible only when there is an appropriate injection of energy, and a change of entropy moves the system in a desired direction (backward and forward). Looking to biological systems where backward and forward evolutions depend on specific physical conditions, we present a controlled reversibility in a bio-inspired computing. This means that we may have both reversible and irreversible steps, where the reversible steps are triggered by a special rollback symbol ρ. Specific sequences of this special symbol coming from the environment allow to control the direction of the computation in order to recover from failures or to avoid deadlocks. This idea is somehow similar to that presented in [Mezzina and Tuosto 2017] where evolution branches in a global graph are decorated with so-called reversion guards, namely conditions on the state of the system triggering a backward computation. The computation proceeds forward until the guard of the branch becomes true, a moment when the computation is reverted in order to find a better branch to execute (if such a branch exists).

We describe such a controlled reversibility in parallel rewriting systems over multisets, a computation model used by some bio-inspired formalisms. Parallel rewriting systems over multisets (PRS) consist of a set of rewriting rules over multisets of objects, together with an initial multiset of objects; PRS are used to describe the dynamics of systems which involve parallel access to resources [Bistarelli et al. 2003]. The evolution of a PRS consists of applying rules over available resources (objects) in a maximal parallel rewriting manner. PRS can represent directly some variants of membrane systems [Păun 2002, Păun et al. 2010], as well as some variants of Petri nets [Reisig 1985] by representing transitions as rules and places as object names. The difference between membrane systems and Petri nets as PRS is given by the evolution strategy; the strategy used in membrane systems is given by a maximally parallel application of rules, while in Petri nets we have in general an unconditional application of transitions. In order to exemplify our approach, we present a controlled reversibility in membrane systems.

The structure of the paper is as follows. Section 2 presents briefly the parallel rewriting systems over multisets. Section 3 shows how parallel rewriting systems can represent directly some variants of membrane systems as well as some variants of Petri nets. Section 4 represents the main part of the paper; it presents the reversible membrane systems and some important properties of these systems. Conclusion and references end the paper.

2 Parallel Rewriting Systems over Multisets

A multiset w over a set X is a function $w : X \to \mathbb{N}$ from the set X to the set of natural numbers $\mathbb{N}$. The multiset w is finite if the support $sup(w) = \{a \in X \mid w(a) \neq 0\}$ is finite. Throughout this paper we consider finite multisets, unless specifically mentioned otherwise. When describing a multiset characterized by having the support $\{a, b, c, d\}$ with $w(a) = 3$, $w(b) = 4$, $w(c) = 1$, $w(d) = 2$, we use the simpler representation $3a + 4b + c + 2d$. In particular, elements with multiplicity equal to 1 do not have their multiplicity specified, as is the case of c

in $3a + 4b + c + 2d$. For simplicity, sometimes we overload the set notation to multisets by using $a \in w$ instead of $w(a) \geq 1$. A multiset w is called non-empty if it contains at least one element, namely $|supp(w)| > 0$. We denote the empty multiset by λ.

The sum of two multisets w, w' over X is the multiset $w+w' : X \to \mathbb{N}$ defined by $(w + w')(a) = w(a) + w'(a)$. For two multisets w, w' over X we say that w is contained in w' if $w(a) \leq w'(a)$ for all $a \in X$, and we denote this by $w \leq w'$. If $w \leq w'$, we can define $w' - w$ by $(w' - w)(a) = w'(a) - w(a)$.

Formally, a parallel rewriting system over multisets is a tuple $(O, \mathcal{R}, w_0)$ consisting of an alphabet of objects O, a set of rules $\mathcal{R}$ and an initial multiset of objects w_0. Each rule $r \in \mathcal{R}$ has two associated non-empty multisets over O denoted by $lhs(r)$ and $rhs(r)$. The two multisets are called the left-hand side and the right-hand side of the rule. The notation employed for describing a rule together with its left and right-hand sides is $r : lhs(r) \to rhs(r)$. When considering a multiset F of rules, we extend the notations for left-hand side and right-hand side to the entire multiset: $lhs(F) = \sum_{r\in\mathcal{R}} F(r)\cdot lhs(r)$, and similarly $rhs(F) = \sum_{r\in\mathcal{R}} F(r) \cdot rhs(r)$.

A parallel rewriting system $(O, \mathcal{R}, w_0)$ evolves by applying a multiset of rules to the initial multiset, then applying yet another multiset of rules to the multiset obtained from the first application and so on, possibly imposing certain restrictions on the multisets of rules that are applied. A rule $r : u \to v$ can be applied to a multiset w of objects if $lhs(w) \leq w$. The application of the rule r produces the multiset of objects $w - lhs(r) + rhs(r)$ which is obtained by subtracting the left-hand side u from w and adding the right-hand side v. A multiset R of rules can be applied to a multiset of objects w if $lhs(R) \leq w$; the result of the application is $w - lhs(R) + rhs(R)$. The notion of application (of non-empty multisets of rules) yields a labelled transition system over multisets of objects having multisets of rules as labels. In more details, w' is obtained from w by applying R (denoted by $w \xrightarrow{R} w'$) if R can be applied to w, R is not empty and $w' = w - lhs(R) + rhs(R)$. The non-empty condition ensures that in each transition, at least one rule is applied.

Example 1. Consider the parallel rewriting system $(O, \mathcal{R}, w_0)$ with $O=\{a, \dots, e\}$, $w_0 = a + c$ and $\mathcal{R} = \{r_1, r_2, r_3\}$, where $r_1 : a \to d$, $r_2 : c \to e$, $r_3 : c + d \to b$. Then $a + c \xrightarrow{r_1+r_2} d + e$ is the only one possible maximal parallel rewriting step, while

- $a + c \xrightarrow{r_1+r_2} d + e$;
- $a + c \xrightarrow{r_1} c + d \xrightarrow{r_2} d + e$;
- $a + c \xrightarrow{r_2} a + e \xrightarrow{r_1} d + e$;

- $a + c \xrightarrow{r_1} c + d \xrightarrow{r_3} b$

yield all the possible evolutions in this parallel rewriting system. The first three possibilities yield the same result as when the maximal parallel rewriting is applied. On the other hand, the fourth sequence is fundamentally different since it involves the rule r_3, and it is not a maximal parallel rewriting evolution.

3 Membrane Systems and Petri Nets as PRS

Parallel rewriting systems can represent directly several variants of membrane systems, as well as some variants of Petri nets. In what follows we present a general representation for membrane systems, and an example of how a parallel rewriting system can be represented as a Petri net.

The structure μ of a membrane system (also named P system) is represented by a tree structure (with the *skin* as its root), or equivalently by a string of correctly matching parentheses placed in a unique pair of matching parentheses; each pair of matching parentheses corresponds to a membrane. Graphically, a membrane structure is represented by a Venn diagram in which two sets can be either disjoint, or one is the subset of the other. The membranes are labelled in a one-to-one manner.

A *P system* of degree m is $\Pi = (O, \mu, w_1, \dots w_m, \mathcal{R}_1, \dots \mathcal{R}_m, i_o)$, where:

- O is an alphabet of objects;
- μ is a membrane structure with the membranes labelled by natural numbers $1 \dots m$, in a one-to-one manner;
- w_i are multisets over O associated with the regions $1 \dots m$ defined by μ;
- $\mathcal{R}_1, \dots, \mathcal{R}_m$ are finite sets of rules associated with the membranes $1 \dots m$; the rules have the form $u \to v$, where u is a non-empty multiset of objects and v a multiset over messages of the form $(a, here), (a, out), (a, in_j)$;
- i_0 is either a number between 1 and m specifying the output membrane of Π, or it is equal to 0 indicating that the output is the outer region.

We do not use here the notion of output membrane, and so i_0 will be ignored.

Definition 1. The set $\mathcal{M}(\Pi)$ of membranes in a P system Π together with the membrane structure are inductively defined as follows:

- if i is a label and w is a multiset over $O \cup O \times \{out\}$, then $\langle i|w\rangle \in \mathcal{M}(\Pi)$; $\langle i|w\rangle$ is called an *elementary membrane*, and its structure is $\langle\rangle$;

– if i is a label, $M_1, \ldots, M_n \in \mathcal{M}(\Pi)$ have distinct labels $i_1, \ldots, i_n$, each M_k has structure μ_k and w is a multiset over $O \cup O \times \{out\} \cup O \times \{in_{i_1}, \ldots, in_{i_n}\}$, then $\langle i|w; M_1, \ldots, M_n\rangle \in \mathcal{M}(\Pi)$; $\langle i|w; M_1, \ldots, M_n\rangle$ is called a *composite membrane*, and its structure is $\langle \mu_1 \ldots \mu_n\rangle$.

We use the notation $w(M)$ for the multiset of a membrane M.

Definition 2. We say that a multiset of rules $\mathcal{R}$ is *valid* in a membrane M with label i and content $w(M)$ if $lhs(\mathcal{R}) \leq w(M)$. The multiset $\mathcal{R}$ is called *maximally valid* in M if it is valid and there is no other rule r such that $lhs(r) \leq w(M) - lhs(\mathcal{R})$.

As usually in membrane systems, a computation of Π is a (possibly infinite) sequence of configurations $C_0, C_1, \ldots$. Given a configuration $C_k = (w_{k1}, \ldots, w_{kn})$, then the next configuration C_{k+1} is obtained by applying on each multiset w_{ki} a maximally valid multiset of rules from $\mathcal{R}_i$ in a nondeterministic and maximal parallel manner.

Regarding the Petri nets viewed as PRS, we provide an example that is inspired by a Petri net model of an assembly cell for a manufacturing system studied in [Recalde et al. 2004].

Example 2. The parallel rewriting system $\Pi = (O, \mathcal{R}, w_0)$ is described by the alphabet $O = \{m, x, x_f, x_r, y, y_f, y_r, z\}$, the initial multiset $w_0 = 6m + 3x_f + y_f$ and the rules $\mathcal{R} = \{r_1, \ldots, r_5\}$:

– $r_1 : 3m \to x$;

– $r_2 : m \to y$;

– $r_3 : x + x_f \to x_r$;

– $r_4 : y + y_f \to y_r$;

– $r_5 : 2x_r + y_r \to z + 2x_f + y_f$.

The parallel rewriting system Π seen as a Petri net is depicted in Figure 1. It describes the manufacture of pieces of type x, y and z, where pieces of type x and y are used in the production of z. Rule r_1 states that one piece x is produced from three pieces of material m, while rule r_2 states that one piece y is produced from one piece of material m. Rules r_3 and r_4 describe the interface between the production of pieces x, y and that of piece z, namely that one piece x or y is loaded into a free spot x_f or y_f, making them ready (state denoted by x_r or y_r) for further processing. Rule r_5 states that one piece z is produced from two ready pieces x_r and one ready piece y_r; the production of one piece z also frees two spots x_f and one spot y_f. The initial multiset $w_0 = 6m + 3x_f + y_f$ stands for the availability of 6 pieces of material m, 3 free slots x_f and 1 free slot y_f.

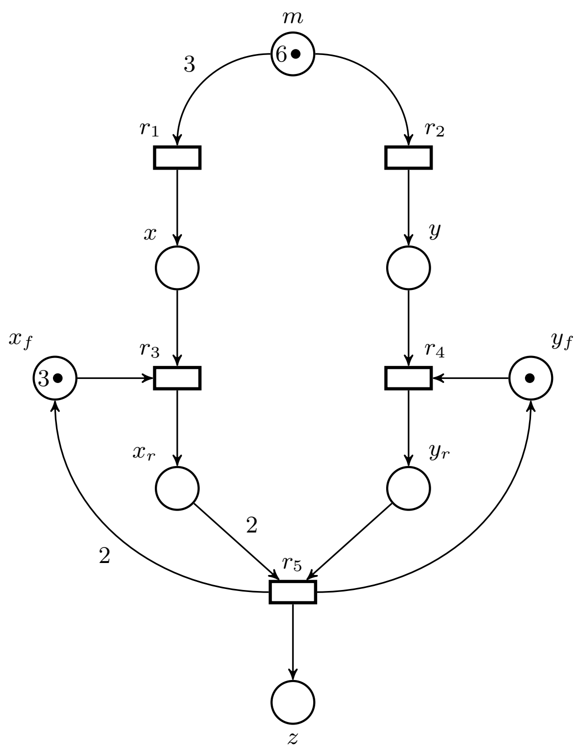

Figure 1: The Petri net corresponding to Π

4 Reversibility in Membrane Systems as PRS

Reversible computation deals with mechanisms for undoing the effects of certain actions executed in a parallel rewriting system. In what follows we are dealing with reversibility in the context of membrane systems viewed as instances of PRS. The key constructions in this investigation are given by adding the reverse rules to the initial set of rules, as well as by adding an external control by means of a special symbol ρ informing the system that a rollback is needed.

In what follows we consider that the parallel rewriting system Π consists of only one compartment labelled by 1 and all rules are of form $u \to v$, with the multisets u and v of objects such that $supp(u), supp(v) \subseteq O$. This means that $\Pi = (O, [\,]_1, \mathcal{R}, w_0)$. In order to reverse the computation of such a system, the most natural approach would be to reverse the rules ($u \to v$ becomes $v \to u$) and to find a condition controlling the evolution of the system (forward or reverse). The control is performed by an active environment which provides at some steps a new distinct object $\rho \notin O$ signalling the system that it has to reverse its computation. This ρ is an abstraction of the physical reality in which a system is informed that a certain change in the environment has an effect on its evolution, as in heat shock response [Voellmy 1994].

We can associate *promoters* and *inhibitors* with a rule $u \to v$ in the form $u \to v|_{w_{prom}}$ and $u \to v|_{\neg w_{inhib}}$ respectively, where w_{prom}, w_{inhib} are non-empty multisets of objects [Bottoni et al. 2002]. A rule $u \to v|_{w_{prom}}$ is applicable only if the objects from w_{prom} are available together with u, while a rule $u \to v|_{\neg w_{inhib}}$ is applicable only if the objects from w_{inhib} are absent while the objects of u are available. It should be noticed that w_{prom} and w_{inhib} are not consumed, but are just used to control the application of the rule $u \to v$. The promoters and inhibitors of membrane systems formalize the reaction enhancing and reaction prohibiting roles of various substances present in cells.

Definition 3. Given a membrane system $\Pi = (O, [\,]_1, \mathcal{R}, w_0)$ of degree 1, a *reversible membrane system* of degree 1 is a tuple $\tilde{\Pi} = (O, [\,]_1, \tilde{\mathcal{R}}, w_0)$, where:

- O, $[\,]_1$ and w_0 are the same alphabet of objects, structure and initial multiset of objects as in the initial system;
- $\tilde{\mathcal{R}} = \overrightarrow{\mathcal{R}} \cup \overleftarrow{\mathcal{R}}_\rho$ is a finite set of rules obtained from the rules of $\mathcal{R}$:
  - for $u \to v \in \mathcal{R}$, we add $u \to v|_{\neg\rho} \in \overrightarrow{\mathcal{R}}$;
  - for $u \to v \in \mathcal{R}$, we add $v \to u|_\rho \in \overleftarrow{\mathcal{R}}_\rho$;
  - $\rho \to \lambda \in \overleftarrow{\mathcal{R}}_\rho$.

We present the evolution of a reversible membrane defined in Definition 3 after providing a simple example of a reversible system.

Example 3. Let us consider the parallel rewriting system $(O, \mathcal{R}, w_0)$ of Example 1, where $\mathcal{R} = \{r_1, r_2, r_3\}$ with $r_1 : a \to d$, $r_2 : c \to e$, $r_3 : c + d \to b$. Applying Definition 3, we obtain the system $\tilde{\Pi} = (O, [\,]_1, \tilde{\mathcal{R}}, w_0)$, where $\tilde{\mathcal{R}} = \overrightarrow{\mathcal{R}} \cup \overleftarrow{\mathcal{R}}_\rho$ represents the set of rules obtained from the rules of $\mathcal{R}$:

- for $r_1 : a \to d \in \mathcal{R}$, we add $\overrightarrow{r_1} : a \to d|_{\neg\rho} \in \overrightarrow{\mathcal{R}}$;
- for $r_2 : c \to e \in \mathcal{R}$, we add $\overrightarrow{r_2} : c \to e|_{\neg\rho} \in \overrightarrow{\mathcal{R}}$;
- for $r_3 : c + d \to b \in \mathcal{R}$, we add $\overrightarrow{r_3} : c + d \to b|_{\neg\rho} \in \overrightarrow{\mathcal{R}}$;
- for $r_1 : a \to d \in \mathcal{R}$, we add $d \to a|_\rho \in \overleftarrow{\mathcal{R}_\rho}$;
- for $r_2 : c \to e \in \mathcal{R}$, we add $\overleftarrow{r_2} : e \to c|_\rho \in \overleftarrow{\mathcal{R}_\rho}$;
- for $r_3 : c + d \to b \in \mathcal{R}$, we add $\overleftarrow{r_3} : b \to c + d|_\rho \in \overleftarrow{\mathcal{R}_\rho}$;
- $\rho \to \lambda \in \overleftarrow{\mathcal{R}}_\rho$.

The biochemical reactions can be seen as causal consequences in which the input causes the output, sometimes under some external influence (e.g., temperature). The principle of causality implies a certain temporal order between some states, and by which any later event is determined by the earlier one. A deep study of causality in parallel rewriting systems over multisets is presented in [Agrigoroaiei and Ciobanu 2014]. We can observe here that the actual time elapsed between the occurrence of consecutive events that are in a given causality relation is not important. Based on these considerations, and inspired by the scenario-based P systems [Ciobanu and Sburlan 2013], we use the regular expressions to define scenarios as a method to model different possibilities of a membrane system.

A regular expression over an alphabet V is defined as follows: (i) λ and each $a \in V$ is a regular expression, (ii) if E_1, E_2 are regular expressions over V, then $E_1; E_2$ and $(E_1)^*$ are regular expressions over V, and (iii) nothing else is a regular expression over V.

A word $c_0; c_1; \ldots; c_n$ (described by a regular expression over the set $\{\lambda; \rho\}$) is called a scenario, and it controls which rules of $\tilde{\mathcal{R}}$ are applied. Given a multiset of objects, if $c_i = \lambda$ then a forward computation takes place by applying the rules from $\overrightarrow{\mathcal{R}}$. On the other hand, if $c_j = \rho$ then a backward computation will take place by applying the rules from $\overleftarrow{\mathcal{R}}_\rho$.

Remark. For the purpose of this paper it is enough to consider the controlled reversibility by considering scenarios using only symbols from $\{\lambda; \rho\}$. The case of considering scenarios using symbols from $\{\lambda; \rho\} \cup O$ represents a further work.

For a scenario $c_0; c_1; \ldots; c_n$ that is started at the same time when the computation starts in $\tilde{\Pi}$ with the multiset w_0, a sequence $W_0; \ldots; W_n$ denotes the evolution of $\tilde{\Pi}$; when $W_0 = w_0 c_0$ represents the initial multiset together with the initial symbol indicating the direction of computation, then the evolution is given by $W_i = w_i c_i$ for all $1 \leq i \leq n$, where $w_{i-1} \xrightarrow{\mathcal{R}_{i-1}} w_i$ for a maximally valid multiset $\mathcal{R}_{i-1}$ with rules from $\tilde{\mathcal{R}}$. A started scenario is said to be entirely applied if all of its symbols are consumed in the given order for consecutive configurations. In the case that there does not exist a maximally valid multiset, the started scenario is said to be interrupted.

We claim that our reversible membrane systems is only a decoration of a membrane system. In fact, as for the most of the existing reversible calculi, such decorations can be erased. It is enough to forget about backward rules by removing the ρ object and scenarios in the reversible membrane systems; in this way, there is no object ρ coming from the environment to inhibit the forward rules. This is formally stated in what follows; the next result proves that a step in the initial membrane system can be modelled by a step in the reversible membrane system by applying only rules from the forward set of rules.

Proposition 4. $w \xrightarrow{\mathcal{R}} w'$ *if and only if* $w \xrightarrow{\overrightarrow{\mathcal{R}}} w'$.

Proof. If $w \xrightarrow{\mathcal{R}} w'$, then $\mathcal{R}$ is valid in w and $w' = w - lhr(\mathcal{R}) + rhs(\mathcal{R})$. By Definition 3, for each rule $r : u \to v \in \mathcal{R}$, there exists a corresponding rule $\overrightarrow{r} : u \to v|_{\neg\rho} \in \overrightarrow{\mathcal{R}}$. This means that $lhs(r) = lhs(\overrightarrow{r})$ and $rhs(r) = rhs(\overrightarrow{r})$, and thus $lhs(\mathcal{R}) = lhs(\overrightarrow{\mathcal{R}})$ and $rhs(\mathcal{R}) = rhs(\overrightarrow{\mathcal{R}})$. This implies that $w' = w - lhs(\mathcal{R}) + rhs(\mathcal{R}) = w - lhs(\overrightarrow{\mathcal{R}}) + rhs(\overrightarrow{\mathcal{R}})$, and thus $w \xrightarrow{\overrightarrow{\mathcal{R}}} w'$ holds.

The other implication is proven in a similar manner. □

If we have two valid multisets of rules that do not compete for the same objects, then the following two results hold.

Proposition 5 (forward diamond). *If* $w \xrightarrow{\overrightarrow{\mathcal{R}}} w'$ *and* $w \xrightarrow{\overrightarrow{\mathcal{R}'}} w''$, *where* $\overrightarrow{\mathcal{R}}$ *and* $\overrightarrow{\mathcal{R}'}$ *are two valid multisets of rules such that* $lhs(\overrightarrow{\mathcal{R}}) \cap lhs(\overrightarrow{\mathcal{R}'}) = \emptyset$, *then there exists a multiset* w_1 *such that* $w' \xrightarrow{\overrightarrow{\mathcal{R}'}} w_1$ *and* $w'' \xrightarrow{\overrightarrow{\mathcal{R}}} w_1$.

Proof. If $w \xrightarrow{\overrightarrow{\mathcal{R}}} w'$ and $w \xrightarrow{\overrightarrow{\mathcal{R}'}} w''$, then $w' = w - lhr(\overrightarrow{\mathcal{R}}) + rhs(\overrightarrow{\mathcal{R}})$ and $w'' = w - lhr(\overrightarrow{\mathcal{R}'}) + rhs(\overrightarrow{\mathcal{R}'})$, respectively. From $lhs(\overrightarrow{\mathcal{R}}) \cap lhs(\overrightarrow{\mathcal{R}'}) = \emptyset$ it follows that the two multisets of rules can both be applied simultaneously on w without overlapping on the available objects, and thus there exists a w_1 such that $w \xrightarrow{\overrightarrow{\mathcal{R}} \cup \overrightarrow{\mathcal{R}'}} w_1$, where $w_1 = w - lhr(\overrightarrow{\mathcal{R}} \cup \overrightarrow{\mathcal{R}'}) + rhs(\overrightarrow{\mathcal{R}} \cup \overrightarrow{\mathcal{R}'})$. It should be noticed that $w_1 = w - lhr(\overrightarrow{\mathcal{R}}) - lhs(\overrightarrow{\mathcal{R}'}) + rhs(\overrightarrow{\mathcal{R}}) + rhs(\overrightarrow{\mathcal{R}'}) = (w - lhr(\overrightarrow{\mathcal{R}}) + rhs(\overrightarrow{\mathcal{R}})) - lhs(\overrightarrow{\mathcal{R}'}) + rhs(\overrightarrow{\mathcal{R}'}) = w' - lhs(\overrightarrow{\mathcal{R}'}) + rhs(\overrightarrow{\mathcal{R}'})$, and thus $w' \xrightarrow{\overrightarrow{\mathcal{R}'}} w_1$. In a similar manner, $w_1 = w - lhr(\overrightarrow{\mathcal{R}}) - lhs(\overrightarrow{\mathcal{R}'}) + rhs(\overrightarrow{\mathcal{R}}) + rhs(\overrightarrow{\mathcal{R}'}) = (w - lhr(\overrightarrow{\mathcal{R}'}) + rhs(\overrightarrow{\mathcal{R}'})) - lhs(\overrightarrow{\mathcal{R}}) + rhs(\overrightarrow{\mathcal{R}}) = w'' - lhs(\overrightarrow{\mathcal{R}}) + rhs(\overrightarrow{\mathcal{R}})$, and thus $w'' \xrightarrow{\overrightarrow{\mathcal{R}}} w_1$. □

Proposition 6 (reverse diamond). *If* $w \xrightarrow{\overleftarrow{\mathcal{R}_\rho}} w'$ *and* $w \xrightarrow{\overleftarrow{\mathcal{R}'_\rho}} w''$, *where* $\overleftarrow{\mathcal{R}_\rho}$ *and* $\overleftarrow{\mathcal{R}'_\rho}$ *are two valid multisets of rules such that* $lhs(\overleftarrow{\mathcal{R}_\rho}) \cap lhs(\overleftarrow{\mathcal{R}'_\rho}) = \emptyset$, *then there exists a multiset* w_1 *such that* $w' \xrightarrow{\overleftarrow{\mathcal{R}'_\rho}} w_1$ *and* $w'' \xrightarrow{\overleftarrow{\mathcal{R}_\rho}} w_1$.

Proof. The proof is similar to that of Proposition 5. □

Now we can prove that if a reversible membrane system performs a forward step by using only rules from $\overrightarrow{\mathcal{R}}$, then it can be matched by a backward step performed by using only rules from $\overleftarrow{\mathcal{R}_\rho}$, and vice-versa. This result is similar to the so-called loop lemma usually proven when studying a reversible formalism [Phillips and Ulidowski 2007].

Proposition 7 (Loop). $w \xrightarrow{\overrightarrow{\mathcal{R}}} w'$ *if and only if* $\rho w' \xrightarrow{\overleftarrow{\mathcal{R}_\rho}} w$.

Proof. If $w \xrightarrow{\overrightarrow{\mathcal{R}}} w'$, then $\overrightarrow{\mathcal{R}}$ is valid in w and $w' = w - lhr(\overrightarrow{\mathcal{R}}) + rhs(\overrightarrow{\mathcal{R}})$. By Definition 3, for each rule $r : u \to v \in \mathcal{R}$, there exist the corresponding rules $\overrightarrow{r} : u \to v|_{\neg\rho} \in \overrightarrow{\mathcal{R}}$ and $\overleftarrow{r} : v \to u|_{\rho} \in \overleftarrow{\mathcal{R}_{\rho}}$. Also, a fresh rule $\rho \to \lambda$ is added to $\overleftarrow{\mathcal{R}}_{\rho}$. This means that $lhs(\overrightarrow{r}) = rhs(\overleftarrow{r})$ and $rhs(\overrightarrow{r}) = lhs(\overleftarrow{r})$, and so $lhs(\overrightarrow{\mathcal{R}}) = rhs(\overleftarrow{\mathcal{R}_{\rho}})$ and $rhs(\overrightarrow{\mathcal{R}}) = lhs(\overleftarrow{\mathcal{R}_{\rho}}) - \{\rho\}$. This implies that $w' = w - lhs(\overrightarrow{\mathcal{R}}) + rhs(\overrightarrow{\mathcal{R}}) = w - rhs(\overleftarrow{\mathcal{R}_{\rho}} + lhs(\overleftarrow{\mathcal{R}_{\rho}}) - \{\rho\}$, and it follows that $w = w' + \{\rho\} - lhs(\overleftarrow{\mathcal{R}_{\rho}}) + rhs(\overleftarrow{\mathcal{R}_{\rho}} = \rho w' - lhs(\overleftarrow{\mathcal{R}_{\rho}}) + rhs(\overleftarrow{\mathcal{R}_{\rho}})$, and thus $\rho w' \xrightarrow{\overleftarrow{\mathcal{R}_{\rho}}} w$ holds.

The other implication is proven in a similar manner. □

Depending on the form of the scenarios, we can get other interesting results.

Proposition 8. *For a membrane system $\tilde{\Pi}$ obtained from a system Π, if the scenario used is λ^*, then the two system have the same sets of configurations.*

Proof. The proof is based on induction on all the maximal valid multisets of rules existing for each configuration, and on Proposition 4 claiming that every forward move of a membrane system can be matched by a forward move of its reversible membrane system, and vice-versa. □

Proposition 9. *For a membrane system $\tilde{\Pi}$ obtained from a system Π, if the scenario used is $(\lambda;\rho)^*$ or $(\rho;\lambda)^*$, then there exists at least one computation such that $w_{2*k} = w_0$, for $k > 1$.*

Proof. If $\mathcal{R}$ is a maximal valid multiset of rules for w_0 in Π and the scenario used is $(\lambda;\rho)^*$, then we may have $w_0 \xrightarrow{\overrightarrow{\mathcal{R}}} w_1$ and $\rho w_1 \xrightarrow{\overleftarrow{\mathcal{R}_{\rho}}} w_2$. Since $w_0 \xrightarrow{\overrightarrow{\mathcal{R}}} w_1$, then due to Proposition 7 we get that $\rho w_1 \xrightarrow{\overleftarrow{\mathcal{R}_{\rho}}} w_0$, and so $w_2 = w_0$. Hence, by induction, the application of the sequence of multisets of rules $(\overrightarrow{\mathcal{R}};\overleftarrow{\mathcal{R}_{\rho}})^*$ leads to an evolution such that $w_{2*k} = w_0$, for $k > 1$.

The proof is similar when the scenario used is $(\rho;\lambda)^*$. □

Proposition 10. *For a membrane system $\tilde{\Pi}$ obtained from a system Π, if the scenario used is $(\lambda^n;\rho^n)^*$ or $(\rho^n;\lambda^n)^*$, then there exists at least one computation such that $w_{2*k*n} = w_0$, for $k > 1$.*

Proof. If $\mathcal{R}_i$ (for $0 \leq i < n$) is a maximal valid multiset of rules for w_i in Π and the scenario used is $(\lambda^n;\rho^n)^*$, we may have $w_0 \xrightarrow{\overrightarrow{\mathcal{R}_0}} w_1 \xrightarrow{\overrightarrow{\mathcal{R}_1}} w_2 \ldots w_{n-1} \xrightarrow{\overrightarrow{\mathcal{R}_{n-1}}} w_n$ and $\rho w_n \xrightarrow{\overleftarrow{\mathcal{R}_{n-1\rho}}} w_{n+1}, \ldots, \rho w_{2n-1} \xrightarrow{\overleftarrow{\mathcal{R}_{n0\rho}}} w_{2n}$. Since $w_0 \xrightarrow{\overrightarrow{\mathcal{R}_0}} w_1 \xrightarrow{\overrightarrow{\mathcal{R}_1}} w_2 \ldots w_{n-1} \xrightarrow{\overrightarrow{\mathcal{R}_{n-1}}} w_n$, then due to Proposition 7 we got that $\rho w_n \xrightarrow{\overleftarrow{\mathcal{R}_{n-1\rho}}} w_{n-1}$, $\ldots$, $\rho w_1 \xrightarrow{\overleftarrow{\mathcal{R}_{n0\rho}}} w_0$, and thus $w_{2n} = w_0$. Hence, by induction, the application of the sequence of multisets of rules $(\overrightarrow{\mathcal{R}_0}; \dots ; \overrightarrow{\mathcal{R}_n}; \overleftarrow{\mathcal{R}_{n\rho}}; \dots ; \overleftarrow{\mathcal{R}_{0\rho}})^*$ leads to an evolution such that $w_{2*k*n} = w_0$, for $k > 1$.

The proof is similar when the scenario is $(\rho^n; \lambda^n)^*$. $\square$

5 Conclusion

Reversibility seems to be a hot topics nowadays, and an important property of computational systems. It has been intensely studied for Turing machines [Bennett 1973, Morita and Yamaguchi 2007], register machines [Morita 1996], cellular automata [Morita 2007], circuits of logical elements [Fredkin and Toffoli 1982] and circuits of memory elements [Morita 2001].

Reversible membrane systems were considered in [Leporati et al. 2006], but the model does not uses maximal parallel rewriting and the main result is the simulation of the Fredkin gate (and thus it studies the reversible circuits). In [Alhazov and Morita 2010] it is studied the reversibility of P systems with maximal parallelism systems only from a computability point of view. The so-called *dual P systems* [Agrigoroaiei and Ciobanu 2008] present reversibility in P systems under the influence of category theory (reversibility as duality).

The approach presented in this paper deals with a controlled reversibility in a more general framework provided by parallel rewriting system over multisets. In particular, we defined and studied reversibility in the context of membrane systems as instances of parallel rewriting systems over multisets. The important innovations of this approach are given by adding the reverse rules to the initial set of rules, as well as by adding an external control by means of scenarios specified by using a special symbol ρ informing the system that a rollback is needed. Several results relating the evolutions of the membrane systems and the reversible membrane systems are presented. As far as we know, it is the first work in this direction, and so it opens new research opportunities.

This approach is different from the constructions needed for reversibility in process calculi [Danos and Krivine 2004] which requires one to assign to each running process an individual memory stack that also serves as a naming scheme and yields a unique identifier for the process. When a forward transition is performed, the information needed for a potential rollback is pushed on the individual memory. In [Phillips and Ulidowski 2007], the structure of the evolving processes is not destroyed, and the progress is marked by underlining the actions that have been performed.

Acknowledgement

This work was partially supported by “A.I.Cuza” University, Faculty of Computer Science, Iaşi, România, and by the COST Action IC1405.

References

[Agrigoroaiei and Ciobanu 2008] Agrigoroaiei, O., Ciobanu, G.: Dual P Systems; Lecture Notes in Computer Science 5391 (2009), 95–107.

[Agrigoroaiei and Ciobanu 2014] Agrigoroaiei, O., Ciobanu, G.: Rewriting Systems Over Indexed Multisets; Computer Journal 57 (2014), 165–179.

[Alhazov and Morita 2010] Alhazov, A., Morita, K.: On Reversibility and Determinism in P System; Lecture Notes in Computer Science 5957 (2010), 158–168.

[Bennett 1973] Bennett, C.H.: Logical Reversibility of Computation; IBM Journal of Research and Development 17 (1973), 525–532.

[Bistarelli et al. 2003] Bistarelli, S., Cervesato, I., Lenzini, G., Marangoni, R. Martinelli, F.: On Representing Biological Systems Through Multiset Rewriting; Lecture Notes in Computer Science 2809 (2003), 415–426.

[Bottoni et al. 2002] Bottoni, P., Martín-Vide, C., Păun, Gh., Rozenberg, G.: Membrane Systems With Promoters/Inhibitors; Acta Informatica 38 (2002), 695–720.

[Cardelli and Laneve 2011] Cardelli, L., Laneve, C.: Reversibility in Massive Concurrent Systems; Scientific Annals of Computer Science 21 (2011), 175–198.

[Ciobanu and Sburlan 2013] Ciobanu, G., Sburlan, D.: Scenario Based P Systems; International Journal of Unconventional Computing 9 (2013), 351–366.

[Danos and Krivine 2004] Danos, V., Krivine, J.: Reversible Communicating Systems; Lecture Notes in Computer Science 3170 (2004), 292–307.

[Fredkin and Toffoli 1982] Fredkin, E., Toffoli, T.: Conservative Logic; International Journal of Theoretical Physics 21 (1982), 219–253.

[Kuhn and Ulidowski 2016] Kuhn, S., Ulidowski, I.: A Calculus For Local Reversibility; Lecture Notes in Computer Science 9720 (2016), 20–35.

[Leporati et al. 2006] Leporati, A., Zandron, C., Mauri, G.: Reversible P Systems to Simulate Fredkin Circuits; Fundamenta Informaticae 74 (2006), 529–548.

[Mezzina and Tuosto 2017] Mezzina, C.A., Tuosto, E.: Choreographies for Automatic Recovery; arXiv:1705.09525 (2017).

[Morita 1996] Morita, K.: Universality of a Reversible Two-Counter Machine; Theoretical Computer Science 168 (1996), 303–320.

[Morita 2001] Morita, K.: A Simple Reversible Logic Element and Cellular Automata for Reversible Computing; Lecture Notes in Computer Science 2055 (2001), 102–113.

[Morita 2007] Morita, K.: Simple Universal One-Dimensional Reversible Cellular Automata; Journal of Cellular Automata 2 (2007), 159–165.

[Morita and Yamaguchi 2007] Morita, K., Yamaguchi, Y.: A Universal Reversible Turing Machine; Lecture Notes in Computer Science 4664 (2007), 90–98.

[Păun 2002] Păun, Gh.: "Membrane Computing. An Introduction"; Springer (2002).

[Păun et al. 2010] Păun, Gh., Rozenberg, G., Salomaa, A. (eds.): "Handbook of Membrane Computing"; Oxford University Press (2010).

[Phillips and Ulidowski 2007] Phillips, I., Ulidowski, I.: Reversing Algebraic Process; Journal of Logic and Algebraic Programming 73 (2007), 70–96 (2007).

[Recalde et al. 2004] Recalde, L., Silva, M., Ezpeleta, J., Teruel, E.: Petri Nets and Manufacturing Systems: An Examples-Driven Tour; Lectures on Concurrency and Petri Nets 3096 (2004), 742–788.

[Reisig 1985] Reisig, W.: "Petri Nets: An Introduction"; Springer (1985).

[Voellmy 1994] Voellmy, R.: Transduction of the Stress Signal And Mechanisms of Transcriptional Regulation of Heat Shock/Stress Protein Gene Expression in Higher Eukaryotes; Critical Reviews in Eukaryotic Gene Expression 4 (1994), 357–401.