P automata with priorities working in sequential and maximally parallel mode *

Luděk CIENCIALA

Institute of Computer Science, Silesian University,
746 01 Opava, Czech Republic
E–mail:ludek.cienciala@fpf.slu.cz

Abstract

In this paper we show direct transformations between P automata with priorities working in sequential and maximally parallel modes.

1 Introduction

P automata were introduced by E. Csuhaj-Varjú and G. Vaszil [2] as a variant of P systems [4]. The structure of P automaton is defined by hierarchically embedded membranes with one outermost membrane which is called the skin membrane. Each membrane can contain some objects and another membranes. With every region a set of communication rules is associated. According to these rules and given priorities among them, the objects evolve and move throught the membranes.

We consider two different ways of applying the rules. If the rules are applied in the sequential manner (one rule per region is used in one step), we speak about P automata working in the sequential mode ([1], [2]). A P automaton working in the maximally parallel mode uses in one step of the computation as much applicable rules as possible ([3]). In this paper we study the relation between the sequential and the maximally parallel way of the rule application.

2 Preliminaries

In this section we review some terminology and definitions. We denote by V^* the set of all words over an alphabet V. The set of natural numbers we denote by N. A multiset of objects M is defined as a pair $M = (V, f)$, where V is the set of objects and f is a mapping $f : V \rightarrow N$. The set V is called a base of the multiset M. We denote by V^0 the set of all multisets with a base V. Every finite multiset M with a base $V = \{a_1, \ldots, a_n\}$ can be represented as a string w over alphabet V which satisfies the condition $|w|_{a_i} = f(a_i)$, where $1 \leq i \leq n$. We denote $supp(M) = \{a \mid a \in V \ \wedge \ f(a) \neq 0\}$. We say that the multiset M is empty if $supp(M) = \emptyset$.

*Research supported in part by the project GAČR 201/04/0528 and by IGS SU č.11/2004.

Let V and $T \subseteq V$ be two alphabets. We define a mapping $h : V^0 \to 2^T$ as $h(M) = supp(M) \cap T$ for a finite multiset $M = (V, f)$; it is called $l - projection$ of V^0 *to* 2^T. We extend the mapping h to a sequence of multisets $M_1, \ldots, M_n,\ M_i \in V^0,\ 1 \leq i \leq n :\ h(M_1 \ldots M_n) = \{x_1 \ldots x_n \mid x_i \in h(M_i)$ for $h(M_i) \neq \emptyset$ and $x_i = \varepsilon$ for $h(M_i) = \emptyset,\ 1 \leq i \leq n\}$.

Let $L_1,\ L_2$ be two languages over multisets $V_1,\ V_2$. We define catenation of languages $L_1 \cdot L_2 = \{M_1 \cup M_2 \mid$ for every $M_1 \in L_1,\ M_2 \in L_2\}$.

A P system (see details in [6]) is a structure of hierarchically embedded membranes. Every membrane has its own label and it delimits a region which can contain objects and other membranes. The skin is always labeled by number 1. A membrane structure is identified by a string of correctly matching parentheses.

Let i and k be two membranes. If the membrane i is contained in the membrane k in the system and there is no other membrane $j,\ j \neq i, k$, such that the membrane i is contained in the membrane j and the membrane j is contained in the membrane k. Then we say that the membrane k is a parent membrane of the membrane i and we use the notation $par(i) = k$. The membrane i is called a child membrane of the membrane k. For each membrane $j \neq 1$ there is membrane $i,\ i < j$ such that $i = par(j)$.

An object is contained in a region if it is not contained in the region of its child membranes.

The objects can be evolved according to rules which are associated with every membrane. By the application of the rules in all regions at the same time the system comes through configurations and pursues a computation.

The system uses rules of the form $(x,\ in)_z\ x,\ z \in V^0$; by the application of this kind of rules in the region i the multiset of objects x is moved from the region $par(i)$ to the region i. This rule can be applicated only if a multiset z called a promoter is contained in the region i. An example of the application of a rule $(ab,\ in)_c$, which is associated with the region 2 is shown in figure 1.

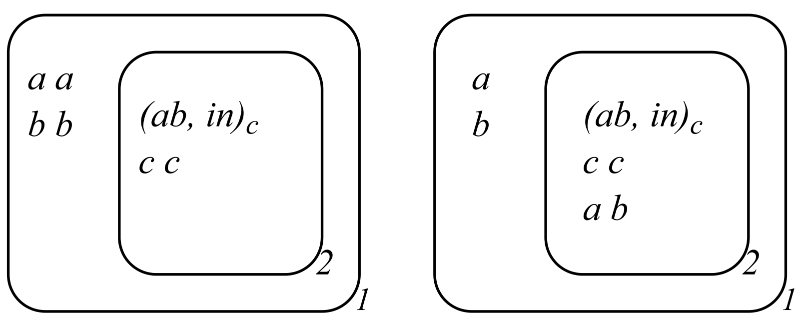

Figure 1: The contents of the regions of the system before and after the application of the rule $(ab,\ in)_c$

The rules can be applied in each region in the sequential or maximally parallel mode. If in one step of a computation in every region only one rule is used and the rule is nondeterministicaly chosen from the set of applicable rules, the system works in sequential mode. If the system works in the maximally parallel mode, then in one step of computation all applicable rules are used.

Definition 1 *An extended P automaton with priorities and with n membranes is a structure:*

$$\Gamma(pri) = (V, T, \mu, (w_1, P_1, R_1, F_1), \ldots, (w_n, P_n, R_n, F_n)), n \geq 1, \text{ where}$$

- V *is an alphabet,*
- $T \subseteq V$ *is a finite set of terminal symbols,*
- μ *is a membrane structure of* n *membranes labeled by* $1, \ldots, n$,
- $w_i \in V^0,\ 1 \leq i \leq n$, *is the initial content of region* i,
- $P_i,\ 1 \leq i \leq n$, *is a finite set of communication rules associated with the region* i*, the communicating rule has a form* $(x,\ in)_z$,
- $R_i \subseteq P_i \times P_i$ *is a relation,* $1 \leq i \leq n$*; if* $(r_1, r_2) \in R_i$*, the rule* r_1 *has higher priority than the rule* r_2 *(we denote* $r_1 > r_2$*). Let* P_1', P_2' *be two sets of rules and* $(P_1', P_2') \in R_i,\ 1 \leq i \leq n$*; then every rule from the set* P_1' *has a higher priority than any rule from the set* P_2',
- $F_i \subseteq V^0,\ 1 \leq i \leq n$, *is the set of final states of region* i, $F_j \neq \emptyset$ *for at least one* $j \neq i$*, where* $1 \leq i \leq n$.

The configuration of P automaton Γ is an n-tuple of multisets of objects contained in the n regions at one moment; $(w_1, \ldots, w_n)$ is the initial configuration of the P automaton Γ. The P automaton Γ is in a final configuration $(x_1, x_2, \ldots, x_n)$ if for all $F_i \neq \emptyset$ there is some $x_i \in F_i,\ 1 \leq i \leq n$.

The end of computation is defined by halting of the system. The P automaton Γ halts if there is no applicable rule and the system is not in a final configuration – this computation is called unsuccessful, or the system is in final configuration – this is a successful computation.

Definition 2 *A transition mapping of an extended P automaton with priorities working in the sequential mode* $\Gamma_{seq}(pri)$ *is a partial mapping* $\delta_{seq} :\ V^0 \times (V^0)^n \to (V^0)^n$*. Let* $(u_1, \ldots, u_n)$, $(u_1', \ldots, u_n')$ *be two configurations and* $u \in V^0$ *be a multiset. We define the transition mapping:*

$$\delta_{seq}(u,\ (u_1, \ldots, u_n)) \ = \ (u_1', \ldots, u_n')$$

iff for every i, $1 \leq i \leq n$, *there is a rule* $(x_i,\ in)_{z_i}$ *fulfilling the following conditions:*

1. $z_i \subseteq u_i$ *for* $z_i \neq \epsilon$*, or* $u_i = \epsilon$ *for* $z_i = \epsilon$,
2. $x_i \subseteq u_{par(i)},\ 2 \leq i \leq n$,

3. $x_1 = u$,

4. *there is no other applicable rule with higher priority fitting conditions 1., 2. and 3.*,

5. $u'_i = u_i \cup x_i - \bigcup_{par(j)=i} x_j$.

We can define the transition mapping of a P automaton working in maximally parallel mode as follows.

Definition 3 *A transition mapping of an extended P automaton with priorities* $\Gamma_{par}(pri)$ *working in maximal parallel mode is a partial mapping* $\delta_{par} : V^0 \times (V^0)^n \to (V^0)^n$. *Let* $(u_1, \ldots, u_n)$, $(u'_1, \ldots, u'_n)$ *be two configurations and* $u \in V^0$ *be a multiset. We define the transition mapping:*

$$\delta_{par}(u,\ (u_1, \ldots, u_n)) \ = \ (u'_1, \ldots, u'_n)$$

iff for every i, $1 \leq i \leq n$ *there exists a multiset of rules* $P'_i = \{r_{i,1}, \ldots, r_{i,m}\}$, *where* $r_{i,j} = (x_{i,j},\ in)_{z_{i,j}} \in P_i$, $1 \leq j \leq m$, *fitting conditions:*

1. $z_{i,j} \subseteq u_i$ *for* $z_{i,j} \neq \epsilon$ *for every* $z_{i,j}$, $1 \leq j \leq m_i$, *or every* $z_{i,j} = \epsilon$, $1 \leq j \leq m_i$, *and* $u_i = \epsilon$,

2. $x_{i,j} \ldots x_{i,m} \subseteq u_{par(i)}$, $2 \leq i \leq n$,

3. $x_{1,1} \ldots x_{1,m} = u$,

4. *there is no other applicable rule with higher priority, which can be used instead of the multiset of rules* P''_i *and* $P''_i \subseteq P'_i$,

5. *there is no other rule* $r_i \in P_i$ *and* $r_i \notin P'_i$ *such that* $\{r_i\} \cup P'_i$ *fulfil the previous conditions*,

6. $u'_i = u_i \cup x_i - \bigcup_{par(j)=i} x_j$.

A sequence of configurations obtained in this manner is called a computation. The computation ends if the system reaches a final configuration, that is, if all regions are in final states. If for some j there is $F_j = \emptyset$, P automaton can enter the final configuration without regard to state of region j, every state of region j is final. The sequence of multisets of objects from T which enter a system during a computation is called the accepted sequence.

Definition 4 *We define an extended transition function* $\overline{\delta_X} : (V^0)^* \times (V^0)^n \to (V^0)^n$, $X \in \{seq, par\}$, *as a function mapping a sequence of multisets over* V *and a configuration* $(u_1, \ldots, u_n)$ *of the P automaton* $\Gamma_X(pri)$ *to a new configuration, if it exists, as follows.*

1. $\overline{\delta_X}(v,\ (u_1, \ldots, u_n)) = \delta_X(v,\ (u_1, \ldots, u_n))$ $v, u_i \in V^0$, $1 \leq i \leq n$,

2. $\overline{\delta_X}(v_1 \dots v_s,\ (u_1, \dots, u_n)) = \delta_X(v_s, \overline{\delta_X}(v_1 \dots v_{s-1},\ (u_1, \dots, u_n)),$
$v_j, u_i \in V^0,\ 1 \le j \le s,\ 1 \le i \le n.$

Now we can define a language accepted by P automaton $\Gamma_X(pri)$:

Definition 5 *Let $\Gamma_X(pri)$ be extended P automaton with priorities. An language accepted by $\Gamma_X(pri)$ is the set of sequences of multisets*

$$L(\Gamma_X(pri)) = \{h(t_1 \dots t_m) \mid \delta_X(t_1 \dots t_n, (w_1, \dots, w_n)) = (v_1, \dots, v_n),$$
$$\text{where } v_j \in F_j \text{ for all } j,\ F_j \ne \emptyset,\ 1 \le j \le n\},$$

where $X \in \{seq, par\}$ and h is the $l-projection$.

3 The sequential mode versus the maximally parallel mode

Theorem 1 *Let $\Gamma_{par}(pri)$ be a P automaton working in the maximally parallel mode and $L(\Gamma_{par}(pri))$ the language accepted by this P automaton. Than there exists a sequential P automaton $\Gamma_{seq}(pri)$ such that $L(\Gamma_{par}(pri)) = L(\Gamma_{seq}(pri))$.*

Proof: Let $\Gamma_{par}(pri) = (V, T, \mu, (w_1, P_1, R_1, F_1), \dots, (w_n, P_n, R_n, F_n))$, $n \ge 2$ be a P automaton with priorities working in the maximally parallel mode. We construct two membranes sequential P automaton $\Gamma_{seq}(pri) = (V', T, (w'_1, P'_1, R'_1, F'_1), (w'_2, P'_2, R'_2, F'_2))$ such that $L(\Gamma_{par}(pri)) = L(\Gamma_{seq}(pri))$.

In the first region of the sequential P automaton there are all symbols contained in all regions of $\Gamma_{par}(pri)$ marked with a superscript which indicates their place in μ. The symbols from the first region are without superscripts and this ensures the equality of accepted languages without the necessity of their further transformation.

For example, let $\Gamma'_{par}(pri)$ be the P automaton with membrane structure $[_1\ [_2\ ]_2\ [_3\ [_4\ ]_4\ ]_3\ ]_1$ and configuration $(ab,\ ac,\ bc,\ aa)$. Then new P automaton $\Gamma'_{seq}(pri)$ has the membrane structure $[_1\ [_2\ ]_2\ ]_1$ and in the first region it contains the multiset of objects $ab\ a^2c^2\ b^3c^3\ a^4a^4$. Figure 2 shows this situation.

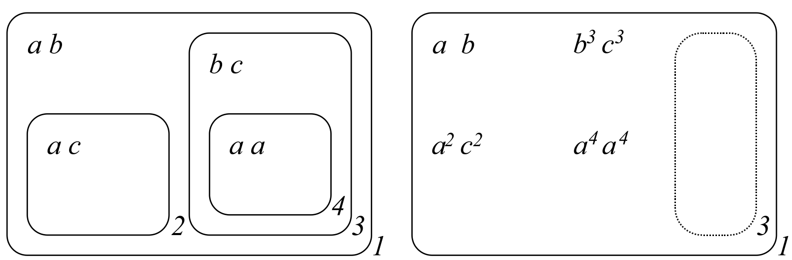

Figure 2: Transformatoin of the contents of the regions

One step of computation of P automaton $\Gamma_{par}(pri)$ is simulated with as many steps of $\Gamma_{seq}(pri)$ as the number of applicable rules are used in this step of computation. If some rule of $\Gamma_{par}(pri)$ is applied to some multiset of objects, rules in

$\Gamma_{seq}(pri)$ move these objects to the second region and objects with a mark of the relevant region and a mark of being new in system enter the system. The entering objects can be used in promoter of another rule in this step of computation of $\Gamma_{par}(pri)$ but they cannot be moved to another membrane. Objects used as a condition for applying some rule are replaced by symbols i_z, where i is the mark of region in which the rule was applied and z is a promoter of the rule.

We define the extended P automaton with priorities and with two membranes as follows:

$$\Gamma_{seq}(pri) = (V', T, (w'_1, P'_1, R'_1, F'_1), (w'_2, P'_2 R'_2, F'_2)), where$$

- $V' = V_1 \cup V_2 \cup V_3 \cup \{F, D, G, P_1, \ldots, P_n, Q_1, \ldots, Q_n, 1, \ldots, n\}$, $F, D, G, P_i, Q_i, i \notin V,\ 1 \le i \le n$,
  - $V_1 = V\ \cup \{x^i | \in V,\ 2 \le i \le n\} \bigcup \{x^{i'}, i_x | x \in V,\ 1 \le i \le n\}$,
  - $V_2 = \{F_y | y \in V_1\}$,
  - $V_3 = \{i'_u | u\ \in V_1^*,\ 1 \le i \le n\}$;
- $w'_1 = w_1 w_2 \ldots w_n FG12 \ldots n$;
- $w'_2 = D$;
- $P'_1 = M_0 \cup M_1 \cup \ldots \cup M_{n+4}$,
  - $M_0 = \{(Q_i,\ in)_{P_i} |\ 1 \le i \le n\}$,
  - $M_1 = \{(1_z 1'_z x P_1 u,\ in)_{zG} |\ (x,\ in)_z \in P_1,\ if\ z = \epsilon\ then\ u = E_1\ else\ u = \epsilon\}$,
  - $M_i = \{((j)'_w x^{i'} i_z i'_z P_i u,\ in)_{w^j z^i G} |\ (x,\ in)_z \in P_i,\ 2 \le i \le n-1$, $w \in \{x^i, i_x | x \in V\}^*, j = par(i), if\ z = \epsilon\ then\ u = E_1\ else\ u = \epsilon\}$,
  - $M_n = \{((par(n))'_w x^{n'} z^{n'} n'_z P_n u,\ in)_{w^{par(n)} z^n G} |\ (x,\ in)_z \in P_n,\ if\ z = \epsilon\ then\ u = E_1\ else\ u = \epsilon\}$,
  - $M_{n+1} = \{(F_{x^i} x_i,\ in)_{x^{i'} F'},\ (F_{i_x} x_i,\ in)_{i'_x F'},\ (F_{1_x} x,\ in)_{1_x F'}$, $(F_x x,\ in)_{x' F'} |\ x \in V\ ,\ 2 \le i \le n\}$,
  - $M_{n+2} = \{(W,\ in)_i |\ 1 \le i \le n\}$,
  - $M_{n+3} = \{F12 \ldots n,\ in)_G\}$,
  - $M_{n+4} = \{(W'W',\ in)_{Q_n n}, (W'W',\ in)_{Q_n}\}$;
- $P'_2 = N_0 \cup N_1 \cup \ldots \cup N_{n+3}$,
  - $N_0 = N_{01} \cup N_{02}$,
    - $N_{01} = \{(Q_i i,\ in)_D |\ 1 \le i \le n\}$,
    - $N_{02} = \{(Q_i,\ in)_D |\ 1 \le i \le n\}$;
  - $N_1 = \{(1'_x P_1,\ in)_D |\ (x,\ in)_z \in P_1\}$,
  - $N_i = \{((par(i))'_w w^{par(i)} i'_z z^i P_i,\ in)_D |\ (x,\ in)_z \in P_i\}$,

- $N_n = \{((par(n))'_w w^{par(n)} n'_z z^n P_n,\ in)_D |\ (x,\ in)_z \in P_n\}$,
- $N_{n+1} = \{(F_{x^i},\ in)_D, (F_{i_x} i_x,\ in)_D, (F_x x',\ in)_D, (W',\ in)_D |$ $x \in V, 1 \leq i \leq n\}$,
- $N_{n+2} = \{(WG,\ in)_D, (F_{x^i} G,\ in)_D, (F_{i_x} G,\ in)_D, F_x G,\ in)_D |$ $x \in V, 1 \leq i \leq n\}$,
- $N_{n+3} = \{(F,\ in)_D\}$,
- $N_{n+4} = \{(E_i G v,\ in)_D |\ v \in \{x^i, i_x |\ x \in V\}\}$;

- $R'_1 = \{M_{n+1} > M_0 > M_1 > \ldots > M_n > M_{n+4} > M_{n+2} > M_{n+3}\}$, priorities among the rules of original P automaton working in maximally parallel mode are preserved in a scope of the sets $M_1, \ldots, M_n$,
- $R'_2 = \{N_{01} > N_{02} > N_{n+4} > N_1 > \ldots > N_{n+3}\}$;
- $F'_1 = T_1 \cdot T_2^2 \cdot \ldots \cdot T_n^n \cdot FG1 \ldots n$, where $T_i^i = F_i^i$ for $F_i \neq \emptyset$,else $T_i^i = (V^i)^*$,
- $F'_2 = \emptyset$.

Consider one step of the P automaton working in maximally parallel mode. The sequential P automaton nonderministically chooses one rule from the set of applicable rules from the set M_1; this ensures that this rule is of the form

$$(1_z 1'_z x' P_1,\ in)_{zG}.$$

For example, if

$$\begin{aligned} z &= a_1 a_2 \ldots a_k, \\ x &= b_1 b_2 \ldots b_l, \end{aligned}$$

then the multiset

$$1_{a_1} 1_{a_2} \ldots 1_{a_k} 1'_{a_1 a_2 \ldots a_k}\ b'_1 b'_2 \ldots b'_l P_1$$

enters the system and in the following step a symbol Q_1 enters the system and the multiset

$$1'_{a_1 a_2 \ldots a_k} a_1 a_2 \ldots a_k P_1$$

leaves the first region. The symbols 1_{a_i} are used to indicate that the symbols a_i, $1 \leq i \leq n$, are included in the promoter of the applied rule. The symbol $1'_{a_1 \ldots a_k}$ determines the symbols which in the next step will leave the first region. By an application of the first rule corresponding with the rule associated with the first region of $\Gamma_{par}(pri)$ the symbol 1 enters the second region. If the symbol 1 is missing in the region 1 it means that at least one rule was applied in the first region of P automaton $\Gamma_{par}(pri)$. At this moment the system is ready to nondeterministically choose further rule from the set M_1. If for the next step of computation there is no applicable rule in the set M_1, the system starts to apply rules from the set M_2. Let the chosen rule be

$$r = (b_1 \ldots b_l,\ in)_{a_1 \ldots a_k}.$$

The subset of the set M_2 is associated with the rule r. The rules are as follows

$$(1'_{U_1\dots U_l} b_1^{2'} \dots b_l^{2'} 2_{a_1} \dots 2_{a_k} 2_{a_1\dots a_k} P_2,\ in)_{U_1\dots U_l a_1^2 \dots a_k^2 G}\ |U_j \in \{b_j, 1_{b_j}\},\ 1 \le j \le l.$$

For example, if in the first region there are symbols

$$a_1^2 a_2^2 \dots a_k^k\ b_1 1_{b_2} 1_{b_3} b_4 \dots b_l,$$

then after using corresponding rule the multiset

$$1'_{b_1 1_{b_2} 1_{b_3} b_4 \dots b_l} b_1^{2'} b_2^{2'} \dots b_n^{2'} 2_{a_1} 2_{a_2} \dots 2_{a_k} 2'_{a_1\dots a_k} P_2$$

moves to the region 2. In next step of computation the symbols i'_{chain} and symbols in *chain* are moved to the region 2 too. The computation continues till the last applicable rule from the set M_n is used. After checking that in the first region there is none of the symbols $1, 2, \dots, n$, the simulation enters to its second part. If there are some of the symbols $1, 2, \dots, n$ in the first region it means that in the simulation of this step of computation of $\Gamma_{par}(pri)$ there is no applicable rule in some region and the computations of both P automata unsuccessfully halt.

In the second part of simulation the system replaces all "used" symbols (symbols i_{a_j}, $a_j^{i'}$ are replaced with a_j^i, $1 \le i, j \le l$). As soon as the last "used" symbol is removed, the simulation of the next step of computation of $\Gamma_{par}(pri)$ can be started.

Every rule $(x,\ in)_\epsilon \in P_i, 1 \le i \le n$, corresponds to the set of rules of type $(wE_iP_i,\ in)_G \in P'_1$ and to the set of rules of type $(vE_iGP_i, in)_D$, which has higher priority than the set of rules of type $(E_iP_iw, in)_D$ both associated to the region 2. If the rule corresponding to the rule $(x, in)_\epsilon$ associated with region i is used then the special symbol E_i enters the system. If there is some symbol with label of region i in the first region, the furhter computation is unsuccessfully halted by taking away the symbol G.

The set of finite states associated with the first region is created by the union of all finite states of $\Gamma_{par}(pri)$ with corresponding label.

If the computation of the P automaton $\Gamma_{par}(pri)$ starts from an initial configuration and ends in a final one and $\Gamma_{par}(pri)$ accepts a word w, then there exists a successful computation of sequential P automaton with two membranes and with priorities $\Gamma_{seq}(pri)$ which accepts the word w too and if $\Gamma_{seq}(pri)$ accepts a word w', then there exists a successful computation of the P automaton $\Gamma_{par}(pri)$, which accepts the word w' too. □

Theorem 2 *Let* $\Gamma_{seq}(pri) = (V, T, \mu, (w_1, P_1, R_1, F_1), \dots, (w_n, P_n, R_n, F_n))$, $n \ge 2$, *be a P automaton with priorities working in the sequential mode and* $L(\Gamma_{seq}(pri))$ *be the language accepted by this P automaton. Then there exists a P automaton* $\Gamma_{par}(pri) = (V', T, \mu, (w'_1, P'_1, R'_1, F'_1), \dots, (w'_n, P'_n, R'_n, F'_n))$ *working in the maximally parallel mode such that* $L(\Gamma_{par}(pri)) = L(\Gamma_{seq}(pri))$.

Sketch of the proof: Let $L(\Gamma_{seq}(pri))$ be the language accepted by the P automaton $\Gamma_{seq}(pri) = (V, T, \mu,\ (w_1, P_1, R_1, F_1), \dots,\ (w_n, P_n, R_n, F_n))$, $n \ge 2$, working in the sequential mode. We will construct the P automaton $\Gamma_{par}(pri) = (V', T, \mu,$ $(w'_1,\ P'_1,\ R'_1,\ F'_1),\ \dots,\ (w'_n, P'_n, R'_n, F'_n))$ such that $L(\Gamma_{par}(pri)) = L(\Gamma_{seq}(pri))$.

The idea of the construction is to add the special symbol r_i $(1 \leq i \leq n)$ to each region and to the promoter of every rule. Then in every region in every step there is only one rule applicable because in each region there is only one occurrence of the special symbol r_i. We need to give a special attention to transformation of the rules of type $(x,\ in)_\varepsilon$. This kind of rules can be applied only if the region is empty. By adding the symbol r_i to the region it cannot be empty anywhere and a modification of the rule $(x,\ in)_\varepsilon$ to the rule $(x,\ in)_{r_i}$ changes the condition for application in undesirable way. This new rule would have to be used only if in the region there is only one symbol and this symbol is r_i. Therefore there is a set of rules $\{D,\ in)_{r_1 u} | if(x,\ in)_\epsilon \in P_1, u \in V\ \}$ associated with region 1 of $\Gamma_{par}(pri)$ with higher priority then the rule $(x,\ in)_{r_1}$. If the rule $(x,\ in)_\epsilon$ is associated with region i, $2 \leq i \leq n$, the set of rules $\{r_{i-1},\ in)_{r_i u}\ u\ \in V\}$ with higher priority than the rule $(x,\ in)_{r_i}$ is associated with region i of P automaton $\Gamma_{par}(pri)$. To every P_i' the rule $(r_{i-1}D,\ in)_{r_i}$ is added and this rule has the highest priority.

The P automaton $\Gamma_{par}(pri)$ accepting the language $L(\Gamma_{seq}(pri))$ is constructed as follows:

$$\Gamma_{par}(pri) = (V', T, \mu, (w_1', P_1', R_1', F_1'), \ldots, (w_n', P_n', R_n', F_n')), where$$

- $V' = V\ \cup \{D, r_i | 1 \leq i \leq n\}$;
- $w_i' = w_i r_i,\ 1 \leq i \leq n$;
- $P_1' = M_1 \cup M_2 \cup M_3$,
  - $M_1 = \{(x,\ in)_{z r_1} |\ (x,\ in)_z \in P_1, z\ \neq \epsilon\}$,
  - $M_2 = \{(x,\ in)_{r_1} |\ (x,\ in)_\epsilon \in P_1\}$,
  - $M_3 = \{D,\ in)_{r_1 u} |\ u \in V, if(x,\ in)_\epsilon \in P_1\}$;
- $P_i' = N_1 \cup N_2 \cup N_3 \cup N_3 \cup N_4, 2 \leq i \leq n$,
  - $N_1 = \{(x,\ in)_{z r_i} |\ (x,\ in)_z \in P_i, z\ \neq \epsilon\}$,
  - $N_2 = \{(x,\ in)_{r_i} |\ (x,\ in)_\epsilon \in P_i\}$,
  - $N_3 = \{(r_{i-1},\ in)_{r_i u} |\ u\ \in V, if(x,\ in)_\epsilon \in P_i\}$,
  - $N_4 = \{D r_{i-1},\ in)_{r_i}\}$;
- $R_1' = \{M_3 > M_2\} \cup R_1$;
- $R_i' = \{M_4 > M_3 > M_2\} \cup R_i,\ 2 \leq i \leq n$;
- $F_i' = F_i \dot{} \{r_i\}, 1 \leq i \leq n$.

Let $\Gamma_{seq}(pri) = (V, T, \mu, (w_1, P_1, R_1, F_1), \ldots, (w_n, P_n, R_n, F_n))$ be a P automaton with priorities working in sequential mode and $(v_1, v_2, \ldots, v_n)$ be its configuration in some step of computation. The corresponding configuration of P automaton $\Gamma_{par}(pri) = (V', T, \mu, (w_1', P_1', R_1', F_1'), \ldots, (w_n', P_n', R_n', F_n'))$ is $(v_1 r_1, v_2 r_2, \ldots, v_n r_n)$. For the next step of computation of $\Gamma_{seq}(pri)$ the n-tuple of rules is nondeterministically chosen from all applicable rules. Then there exists an n-tuple of corresponding rules which has to be applied in the next step of computation of $\Gamma_{par}(pri)$ and after this step the system enters the configuration $(v'_1 r_1, v'_2 r_2, \ldots, v'_n r_n)$ corresponding to a configuration of $\Gamma_{seq}(pri)$ after using n-tuple of chosen rules. The P automaton $\Gamma_{seq}(pri) = (V, T, \mu, (w_1, P_1, R_1, F_1), \ldots, (w_n, P_n, R_n, F_n))$ working in sequential mode accepts the word w iff the P automaton $\Gamma_{par}(pri) = (V', T, \mu, (w'_1, P'_1, R'_1, F'_1), \ldots, (w'_n, P'_n, R'_n, F'_n))$ accepts it too. □

4 Summary

The aim of this paper was to present a direct equivalence of P automata working in the sequential and the maximally parallel mode. We show that for every P automaton working in the maximally parallel mode there exists a P automaton working in the sequential mode such that languages accepted by these P automata are equal. We conclude the paper giving an informal proof of a reverse theorem.

References

[1] Cienciala, L., Ciencialová, L.: P automata with priorities. *Pre-Proceedings of the Workshop on Membrane Computing* (Alhazov, A., Martín-Vide, C., Paun,G.), Tarragona, Spain, 2003, 161–168.

[2] Csuhaj-Varjú, E., Vaszil, G.: P automata. *Pre-Proceedings of Workshop on Membrane Computing (WMC-CdeA2002)*(Paun, Gh., Zandron, C.), Curtea de Arges, Romania, 2002, 177–192.

[3] Csuhaj-Varjú, E., Vaszil, G.: New results and research directions concerning P automata, accepting P systems with communication only. *Report from Brainstroming Week on Membrane Computing*, Tarragona, Spain, 2003, 171–179.

[4] Freud, R., Martín-Vide, C., Obtulowicz, A., Paun, G.: On three classes of automata-like P system. *Proceedings of DTL 2003*, Szeged, Hungary, LNCS 2710, Springer-Verlag, 2003, 292–303.

[5] Paun, Gh.: Computing with membranes. *Turku Centre for Computer Science - TUCS Report No 208*, Turku, Finland, 1998.

[6] Paun, Gh.: *Membrane Computing. An Introduction.* Springer-Verlag, Berlin, Heidelberg, 2002.

[7] P systems web page: http://psystems.disco.unimib.it