A Formal Framework for P Systems

Rudolf Freund[1] Sergey Verlan[2]

[1]Faculty of Informatics, Vienna University of Technology
Favoritenstr. 9, 1040 Vienna, Austria
Email: rudi@emcc.at
[2]LACL, Département Informatique
UFR Sciences et Technologie, Université Paris XII
61, av. Général de Gaulle, 94010 Créteil, France
Email: verlan@univ-paris12.fr

Abstract

The formalism of P systems is known for many years, yet just recently new derivation modes and halting conditions have been proposed. For developing comparable results, a formal description of their functioning, in particular, of the derivation step is necessary. We introduce a formal general framework for static membrane systems that aims to capture most of the essential features of (tissue) P systems and to define their functioning in a formal way.

1 Introduction

P systems were introduced by Gh. Păun (see [8], [14]) as distributed parallel computing devices, based on inspiration from biochemistry, especially with respect to the structure and the functioning of a living cell. The cell is considered as a set of compartments enclosed by membranes; the membranes are nested one in another and contain objects and evolution rules. The basic model neither specifies the nature of these objects nor the nature of the rules. Specifying these two parameters, a lot of different models of computing have been introduced, see [20] for a comprehensive bibliography. Tissue P systems, first considered by Gh. Păun and T. Yokomori in [18] and [19], also see [11], use the graph topology in contrast to the tree topology used in the basic model of P systems.

In this paper, we design a general class of multiset rewriting systems containing, in particular, P systems and tissue P systems. We recall that any P system may be seen at the most abstract level as a multiset rewriting system with only one compartment, encoding the membrane as part of the object representation. However, this approach completely ignores the inner structure of the system because all structural information is hidden (by an encoding) which makes it difficult do deduce any compartment-related information or to model

(processes in) biological systems. At a lower level of abstraction, a P system may be seen as networks of cells (compartments) evolving with multi-cell multi-set rewriting rules. At the lowest level, the graph/tree structure appears as well as a specialization of rules which are of a very particular form. This last level is usually used in the area of P systems because it permits to easily specify the system and to incorporate different new types of rules.

It is worth noting that in the definition of membrane systems the application of rules often is defined in a quite informal way. This is related to the fact that for a long time only the maximally parallel derivation mode was considered and a P system was supposed to work only in this mode. Recent developments in P systems area have revealed that other derivation modes as the minimally parallel derivation mode might be considered [5] and allow for many interesting new results, yet depending on specific interpretations of this notion. Moreover, different halting conditions have been investigated (see [10], [1]), too. All these articles have shown that there is a need for a formal definition of part of the semantics of membrane systems as the derivation step and the halting procedure like it was done for splicing test tube systems [6] or networks of language processors [7]. In particular, this is important for a classification of P systems as well as for their implementation. For approaches to find operational and logic based semantics for P systems we refer to [4] and [2]; a Petri net semantics for membrane systems is discussed in [12].

This article is an attempt to fulfill the goal of formally defining a procedural semantics for a quite large number of well-known variants of P systems considered so far in the literature, but, of course, we do not at all claim to have captured all the variants having already appeared in the literature. In order to be quite general we place our reasoning at the abstract level of *networks of cells*, already considered in a slightly different way in [3]. We adapt an implementational point of view and also give a formal definition of the derivation step, the halting condition and the procedure for obtaining the result of a computation. Moreover, we give examples of applying our concepts to some well-known variants of P systems.

2 Preliminaries

We recall some of the notions and the notations we use (for further details see [8] and [17]). Let V be a (finite) alphabet; then V^* is the set of all strings (a language) over V, and $V^+ = V^* - \{\lambda\}$ where λ denotes the empty string. FIN ($FIN\,(V)$)denotes the set of finite languages (over the alphabet V), and RE, REG, and MAT^{λ} denote the families of recursively enumerable and regular languages as well as matrix languages, respectively. For any family of string languages F, PsF denotes the family of Parikh sets of languages from F and NF the family of Parikh sets of languages from F over a one-letter alphabet. By $\mathbb{N}$ we denote the set of all non-negative integers, by $\mathbb{N}^k$ the set of all vectors of non-negative integers.

Let V be a (finite) set, $V = \{a_1, ..., a_n\}$. Then a finite *multiset* S over V is

a mapping $f_S : V \longrightarrow \mathbb{N}$. The mapping f_S specifies the number of occurrences of each $x \in V$ in S. The size of the multiset S is $|S| = \sum_{x \in V} f_S(x)$. A multiset S over V can also be represented by any string x that contains exactly $f_S(a_i)$ symbols a_i for all $1 \leq i \leq n$, e.g., by $a_1^{f_S(a_1)} ... a_n^{f_S(a_n)}$, or else by the set $\left\{ a_i^{f_S(a_i)} \mid 1 \leq i \leq n \right\}$. The support of S is the set $supp(S) = \{a \in V \mid f(a) \geq 1\}$. For example, the multiset over $\{a, b, c\}$ defined by the mapping $a \to 3, b \to 1, c \to 0$ can be specified by a^3b or $\left\{a^3, b\right\}$, its support is $\{a, b\}$.

The set of all finite multisets over the set V is denoted by $\langle V, \mathbb{N} \rangle$. We may also consider mappings f_S of form the $f_S : V \longrightarrow \mathbb{N}_\infty$ where $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$, i.e., elements of S may have an infinite multiplicity; we shall call such multisets where $f_S(a_i) = \infty$ for at least one i, $1 \leq i \leq n$, *infinite* multisets. The set of all such multisets S over V with $f_S : V \longrightarrow \mathbb{N}_\infty$ is denoted by $\langle V, \mathbb{N}_\infty \rangle$.

Let x and y be two multisets over V, i.e., from $\langle V, \mathbb{N} \rangle$ or $\langle V, \mathbb{N}_\infty \rangle$. Then x is called a submultiset of y, written $x \leq y$ or $x \subseteq y$, if and only if $f_x(a) \leq f_y(a)$ for all $a \in V$; if, moreover, $f_x(a) < f_y(a)$ for some $a \in V$, then x is called a strict multiset of y. Observe that for all $n \in \mathbb{N}$, $n + \infty = \infty$, and $\infty - n = \infty$. The sum of x and y, denoted by $x + y$ or $x \cup y$, is a multiset z such that $f_z(a) = f_x(a) + f_y(a)$ for all $a \in V$. The difference of two multisets x and y, denoted by $x - y$ or $x \setminus y$, provided that $y \subseteq x$, is the multiset z with $f_z(a) = f_x(a) - f_y(a)$ for all $a \in V$. Observe that in the following, when taking the sum or the difference of two multisets x and y from $\langle V, \mathbb{N}_\infty \rangle$, we shall always assume $\{f_x(a), f_y(a)\} \cap \mathbb{N} \neq \emptyset$.

If $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$ are vectors of multisets over V, then $X \leq Y$ if and only if $x_i \subseteq y_i$ for all i, $1 \leq i \leq n$; in the same way, sum and difference of vectors of multisets are defined by taking the sum and the diference, respectively, in each component.

3 Network of Cells

In this section we consider a general framework for describing membrane systems with a static membrane structure. We consider membrane systems as a collection of interacting cells containing multisets of objects [3].

Definition 3.1. A *network of cells of degree* $n \geq 1$ (an *NC of degree* $n \geq 1$, for short) is a construct

$$\Pi = (V, w_1, w_2, \ldots, w_n, R)$$

where

1. V is a *finite alphabet*;
2. $w_i \in \langle V, \mathbb{N}_\infty \rangle$, for all $1 \leq i \leq n$, is the multiset *initially associated to cell* i;
3. R is a finite set of *interaction rules* of the form

$$(X \to Y; P, Q)$$

where $X = (x_1, \ldots, x_n)$, $Y = (y_1, \ldots, y_n)$, $x_i, y_i \in \langle V, \mathbb{N} \rangle$, $1 \leq i \leq n$, are vectors of multisets over V and $P = (p_1, \ldots, p_n)$, $Q = (f_1, \ldots, f_n)$, p_i, f_i, $1 \leq i \leq n$ are finite sets of multisets over V.

We remark that in the definition given above w_i might be an infinite multiset. However, in most of the cases, only one cell, called *the environment*, will contain an infinite multiset. Hence we define $Infinite(\Pi)$ as the vector specifying the symbols with infinite multiplicity. More exactly,

$$Infinite(\Pi) = (inf_1, \ldots, inf_n) \text{ where } inf_i = \{a \in V \mid f_{w_i}(a) = \infty\}, \ 1 \leq i \leq n.$$

Moreover, we define inf'_i, $1 \leq i \leq n$, to be the infinite submultisets of w_i taking into account only the symbols with infinite multiplicity, i.e., $f_{inf'_i}(a) = \infty$ for $f_{w_i}(a) = \infty$ and $f_{inf'_i}(a) = 0$ for $f_{w_i}(a) < \infty$, $a \in V$, as well as w'_i, $1 \leq i \leq n$, to be the finite submultisets of w_i taking into account only the symbols with finite multiplicity, i.e., $f_{w'_i}(a) = 0$ for $f_{w_i}(a) = \infty$ and $f_{w'_i}(a) = f_{w_i}(a)$ for $f_{w_i}(a) < \infty$, $a \in V$.

Remark 3.1. We will also use the notation

$$((x_1, 1) \ldots (x_n, n) \rightarrow (y_1, 1) \ldots (y_n, n) ; (p_1, 1) \ldots (p_n, n), (f_1, 1) \ldots (f_n, n))$$

for a rule $(X \rightarrow Y; P, Q)$. Moreover, if some p_i or f_i is an empty set or some x_i or y_i is equal to the empty multiset, $1 \leq i \leq n$, then we may omit it from the specification of the rule.

A network of cells consists of n cells, numbered from 1 to n, that contain (possibly infinite) multisets of objects over V; initially cell i contains multiset w_i. Cells can interact with each other by means of the rules in R. An interaction rule

$$((x_1, 1) \ldots (x_n, n) \rightarrow (y_1, 1) \ldots (y_n, n) ; (p_1, 1) \ldots (p_n, n), (f_1, 1) \ldots (f_n, n))$$

rewrites objects x_i from cells i into objects y_j in cells j, $1 \leq i, j \leq n$ if cells k, $1 \leq k \leq n$, contain all multisets from p_k and do not contain any multiset from f_k. In other words, the first part of the rule specifies the rewriting of symbols, the second part of the rule specifies permitting conditions and the third part of the rule specifies the forbidding conditions. In the next section we give a precise definition for the application of an interaction rule.

For an interaction rule r of the form above, the set

$$\{i \mid x_i \neq \lambda \text{ or } f_i \neq \emptyset \text{ or } p_i \neq \emptyset \text{ or } y_i \neq \lambda\}$$

induces a relation between the interacting cells. However, this relation need not give rise to a *structure* relation like a tree as in P systems or a graph as in tissue P systems (e.g., see [15] for definitions of P systems and tissue P systems), though most models of membrane systems with a static membrane structure can be seen as special variants of NCs, and moreover, a lot of important features of membrane systems, in particular the derivation step and the halting condition, may be described at the level of NCs.

4 Systems with a Static Structure

In this section we consider networks of cells having a static structure, i.e., the number of cells does not change during the evolution of the system. We first define a transition step and then halting conditions.

Definition 4.1. Consider a network of cells $\Pi = (V, w_1, w_2, \ldots, w_n, R)$. A *configuration* of Π is an n-tuple of finite multisets $C = (u_1, \ldots, u_n)$ satisfying $inf_i \cap u_i = \emptyset$. The *initial configuration* of Π is defined as $C_0 = (w'_1, \ldots, w'_n)$ where w'_i, $1 \leq i \leq n$, are the finite multisets from $\langle V, \mathbb{N} \rangle$ with $f_{w'_i}(a) = 0$ for $f_{w_i}(a) = \infty$ and $f_{w'_i}(a) = f_{w_i}(a)$ for $f_{w_i}(a) < \infty$, $a \in V$.

Definition 4.2. We say that an interaction rule $r = (X \to Y; P, Q)$ is *eligible* for the configuration $C = (u_1, \ldots, u_n)$ if and only if for all i, $1 \leq i \leq n$, we have

- $p_i \subseteq u_i \cup inf'_i$ (p_i is a submultiset of $u_i \cup inf'_i$),
- $f_i \not\subseteq u_i \cup inf'_i$ (f_i is not a submultiset of $u_i \cup inf'_i$), and
- $x_i \subseteq u_i \cup inf'_i$ (x_i is a submultiset of $u_i \cup inf'_i$).

Moreover, we require that $x_j \cap (V - inf_j) \neq \emptyset$ for at least one j, $1 \leq j \leq n$. This last condition ensures that at least one symbol appearing only in a finite number of copies is involved in the rule. The set of all rules eligible for C is denoted by $Eligible(\Pi, C)$.

The marking algorithm. Let $C = (w_1, \ldots, w_n)$ be a configuration of a network of cells Π and let R' be a finite multiset over M with M consisting of the (copies of) rules $r_1, \ldots, r_k$, where each $r_i = (X_i \to Y_i; P_i, Q_i) \in Eligible(\Pi, C)$, $X_i = (x_{i,1}, \ldots, x_{i,n})$, $Y_i = (y_{i,1}, \ldots, y_{i,n})$, $1 \leq i \leq k$. Moreover, set $x'_{i,j}$, $1 \leq j \leq n$, to be the finite multisets from $\langle V, \mathbb{N} \rangle$ with $f_{x'_{i,j}}(a) = 0$ for $a \in inf_j$ and $f_{x'_{i,j}}(a) = f_{x_{i,j}}(a)$ for $a \notin inf_j$, $a \in V$, as well as $y'_{i,j}$, $1 \leq j \leq n$, to be the finite multisets from $\langle V, \mathbb{N} \rangle$ with $f_{y'_{i,j}}(a) = 0$ for $a \in inf_j$ and $f_{y'_{i,j}}(a) = f_{y_{i,j}}(a)$ for $a \notin inf_j$, $a \in V$. Then:

1. consider a vector of multisets $Marked_0(\Pi, C, r_1, \ldots, r_k) = (\lambda, \ldots, \lambda)$ of size n and let $i = 1$;
2. if $X'_i \leq C - Marked_{i-1}(\Pi, C, r_1, \ldots, r_k)$, then set

   $$Marked_i(\Pi, C, r_1, \ldots, r_k) = C - Marked_{i-1}(\Pi, C, r_1, \ldots, r_k) - X'_i,$$

   otherwise, end the algorithm and return **false**;
3. if $i = k$ then end the algorithm and return **true**, otherwise set i to $i + 1$ and return to step 2.

If the marking algorithm returns **true** for the pair (C, R') then we say that the configuration C may be *marked* by R', and we define $Marked(\Pi, C, R') = Marked_k(\Pi, C, r_1, \ldots, r_k)$.

Definition 4.3. Consider a configuration C and $R' \subseteq \mathit{Eligible}\,(\Pi, C)$ (i.e., a multiset of eligible rules). We say that the multiset of rules R' is *applicable* to C if the marking algorithm as described above returns **true** and the configuration $Marked\,(\Pi, C, R')$. The set of all multisets of rules applicable to C is denoted by $\mathit{Applicable}\,(\Pi, C)$.

Definition 4.4. Consider a configuration C and a multiset of rules $R' \in \mathit{Applicable}\,(\Pi, C)$. According to the marking algorithm described above, we define the configuration being the result of *applying* of R' to C as

$$\mathit{Apply}\,(\Pi, C, R') = C - Marked\,(\Pi, C, R') + \Sigma_{1 \leq i \leq k} Y_i'.$$

We remark that $\mathit{Apply}(R', C)$ is again a configuration.

For the specific *derivation modes* to be defined in the following, the selection of multisets of rules applicable to a configuration C may only be a specific subset of $\mathit{Applicable}\,(\Pi, C)$.

Definition 4.5. For the derivation mode ϑ, the selection of multisets of rules applicable to a configuration C is denoted by $\mathit{Applicable}\,(\Pi, C, \vartheta)$.

Definition 4.6. For the *asynchronous* derivation mode ($asyn$),

$$\mathit{Applicable}\,(\Pi, C, asyn) = \mathit{Applicable}\,(\Pi, C)\,,$$

i.e., there are no particular restrictions on the multisets of rules applicable to C.

Definition 4.7. For the *sequential* derivation mode ($sequ$),

$$\mathit{Applicable}\,(\Pi, C, sequ) = \{R' \mid R' \in \mathit{Applicable}\,(\Pi, C) \text{ and } |R'| = 1\}\,,$$

i.e., any multiset of rules $R' \in \mathit{Applicable}\,(\Pi, C, sequ)$ has size 1.

The most important derivation mode considered in the area of P systems from the beginning is the *maximally parallel* derivation mode where we only select multisets of rules R' that are not extensible, i.e., there is no other multiset of rules $R'' \supsetneq R'$ applicable to C.

Definition 4.8. For the *maximally parallel* derivation mode (max),

$$\begin{aligned} \mathit{Applicable}\,(\Pi, C, max) \quad = \quad & \{R' \mid R' \in \mathit{Applicable}\,(\Pi, C) \text{ and there is} \\ & \text{no } R'' \in \mathit{Applicable}\,(\Pi, C) \text{ with } R'' \supsetneq R'\}\,. \end{aligned}$$

A derivation mode closely related to the maximally parallel one, yet not considered so far in the literature is the following one, where we not only demand that the chosen multiset R' is not extensible, but also contains the maximal number of rules among all multisets from $\mathit{Applicable}\,(\Pi, C, max)$:

Definition 4.9. For the *maximal in rules maximally parallel* derivation mode $(max_{rule}max)$,

$$\begin{aligned} \mathit{Applicable}\,(\Pi, C, max_{rule}max) \;=\; & \{R' \mid R' \in \mathit{Applicable}\,(\Pi, C, max) \text{ and} \\ & \text{there is no } R'' \in \mathit{Applicable}\,(\Pi, C, max) \\ & \text{with } |R''| > |R'|\}\,. \end{aligned}$$

In the minimally parallel derivation mode, we need an additional feature for the set of rules R, i.e., we consider a partition of R into disjoint subsets R_1 to R_h. For any set of rules $R' \subseteq R$, let $\|R'\|$ denote the number of sets of rules R_j, $1 \leq j \leq h$, with $R_j \cap R' \neq \emptyset$.

There are several possible interpretations of this minimally parallel derivation mode which in an informal way can be described as applying multisets such that from every set R_j, $1 \leq j \leq h$, at least one rule – if possible – has to be used (e.g., see [5]). We start with the basic variant where in each derivation step we only choose a multiset of rules R' from $\mathit{Applicable}\,(\Pi, C, asyn)$ that cannot be extended to $R'' \in \mathit{Applicable}\,(\Pi, C, asyn)$ with $R'' \supsetneqq R'$ as well as $(R'' - R') \cap R_j \neq \emptyset$ and $R' \cap R_j = \emptyset$ for some j, $1 \leq j \leq h$, i.e., extended by a rule from a set of rules R_j from which no rule has been taken into R'.

Definition 4.10. For the *minimally parallel* derivation mode (min),

$$\begin{aligned} \mathit{Applicable}\,(\Pi, C, min) \;=\; & \{R' \mid R' \in \mathit{Applicable}\,(\Pi, C, asyn) \text{ and} \\ & \text{there is no } R'' \in \mathit{Applicable}\,(\Pi, C, asyn) \\ & \text{with } (R'' - R') \cap R_j \neq \emptyset \\ & \text{and } R' \cap R_j = \emptyset \text{ for some } j,\ 1 \leq j \leq h\}\,. \end{aligned}$$

As in the case of the maximally parallel derivation mode, also for the minimally parallel derivation mode we may choose only multisets of rules with the maximal number of rules thus obtaining the maximal minimally parallel derivation mode:

Definition 4.11. For the *maximal in rules minimally parallel* derivation mode $(max_{rule}min)$,

$$\begin{aligned} \mathit{Applicable}\,(\Pi, C, max_{rule}min) \;=\; & \{R' \mid R' \in \mathit{Applicable}\,(\Pi, C, min) \text{ and} \\ & \text{there is no } R'' \in \mathit{Applicable}\,(\Pi, C, min) \\ & \text{with } |R''| > |R'|\}\,. \end{aligned}$$

In the case of the minimally parallel derivation mode, we have two more very interesting variants of possible interpretations, the first one maximizing the sets of rules involved in a multiset to be applied $(max_{set}min)$, and the second one demanding that all sets of rules that could contribute should contribute $(all_{aset}min)$:

Definition 4.12. For the *maximal in sets minimally parallel* derivation mode $(max_{set}min)$,

$$\begin{aligned} \mathit{Applicable}\,(\Pi, C, max_{set}min) \;=\; & \{R' \mid R' \in \mathit{Applicable}\,(\Pi, C, min) \text{ and} \\ & \text{there is no } R'' \in \mathit{Applicable}\,(\Pi, C, min) \\ & \text{with } \|R''\| > \|R'\|\}\,. \end{aligned}$$

Definition 4.13. For the *using all applicable sets minimally parallel* derivation mode $(all_{aset}min)$,

$$\begin{aligned} \mathit{Applicable}\,(\Pi, C, all_{aset}min) \;=\; & \{R' \mid R' \in \mathit{Applicable}\,(\Pi, C, min) \text{ and} \\ & \text{for all } j,\ 1 \leq j \leq h, \\ & R_j \cap \mathit{Applicable}\,(\Pi, C) \neq \emptyset \\ & \text{implies } R_j \cap R' \neq \emptyset\}\,. \end{aligned}$$

The ideas of taking only multisets of rules involving the maximal number of sets of rules and of taking only multisets of rules involving all sets of rules that can contribute now can also be taken over for the maximally parallel derivation mode:

Definition 4.14. For the *maximal in sets maximally parallel* derivation mode $(max_{set}max)$,

$$\begin{aligned} \mathit{Applicable}\,(\Pi, C, max_{set}max) \;=\; & \{R' \mid R' \in \mathit{Applicable}\,(\Pi, C, max) \text{ and} \\ & \text{there is no } R'' \in \mathit{Applicable}\,(\Pi, C, max) \\ & \text{with } \|R''\| > \|R'\|\}\,. \end{aligned}$$

Definition 4.15. For the *using all applicable sets maximally parallel* derivation mode $(all_{aset}max)$,

$$\begin{aligned} \mathit{Applicable}\,(\Pi, C, all_{aset}max) \;=\; & \{R' \mid R' \in \mathit{Applicable}\,(\Pi, C, max) \text{ and} \\ & \text{for all } j,\ 1 \leq j \leq h, \\ & R_j \cap \mathit{Applicable}\,(\Pi, C) \neq \emptyset \\ & \text{implies } R_j \cap R' \neq \emptyset\}\,. \end{aligned}$$

Finally, we should like to mention that the derivation modes $max_{set}X$ and $all_{aset}X$ with $X \in \{max, min\}$ could be extended by the constraint that a maximal number of rules has to be used, too, thus yielding derivation modes $max_{set}max_{rule}X$ and $all_{aset}max_{rule}X$ with $X \in \{max, min\}$. Demanding to use at least one rule from every set R_j, $1 \leq j \leq h$, would be another option, yet this case will be covered by the variant of partial halting defined in the succeeding subsection when being combined with derivation modes as $max_{set}X$ and $all_{aset}X$ with $X \in \{max, min\}$.

For all the derivation modes defined above, we now can define how to obtain a next configuration from a given one by applying an applicable multiset of rules according to the constraints of the underlying derivation mode:

Definition 4.16. Given a configuration C of Π and a derivation mode ϑ, we may choose a multiset of rules $R' \in \mathit{Applicable}\,(\Pi, C, \vartheta)$ in a non-deterministic way and apply it to C. The result of this *transition step* from the configuration C with applying R' is the configuration $\mathit{Apply}\,(\Pi, C, R')$, and we also write $C \Longrightarrow_{(\Pi,\vartheta)} C'$. The reflexive and transitive closure of the transition relation $\Longrightarrow_{(\Pi,\vartheta)}$ is denoted by $\Longrightarrow^*_{(\Pi,\vartheta)}$.

There are several other derivation modes considered in the literature, e.g., we may apply (at most) k rules in parallel in every derivation step, but we leave the task to define such derivation modes in the general framework elaborated in this paper to the reader.

Definition 4.17. A configuration C is said to be *accessible* in Π with respect to the derivation mode ϑ if and only if $C_0 \Longrightarrow^*_{(\Pi,\vartheta)} C$ (C_0 is the initial configuration of Π). The set of all accessible configurations in Π is denoted by $Accessible\,(\Pi)$.

Definition 4.18. A derivation mode ϑ is said to be *deterministic* (*det-*ϑ) if $|Applicable\,(\Pi, C, \vartheta)| = 1$ for any accessible configuration C.

4.1 Halting conditions

A halting condition is a predicate applied to an accessible configuration. The system halts according to the halting condition if this predicate is true for the current configuration. In such a general way, the notion halting with final state or signal halting can be defined as follows:

Definition 4.19. An accessible configuration C is said to fulfill the *signal halting* condition or *final state halting* condition (S) if and only if

$$S\,(\Pi, \vartheta) \quad = \quad \{C' \mid C' \in Accessible\,(\Pi) \text{ and } State\,(\Pi, C', \vartheta)\}\,.$$

Here $State\,(\Pi, C', \vartheta)$ means a decidable feature of the underlying configuration C', e.g., the occurrence of a specific symbol (signal) in a specific cell.

The most important halting condition used from the beginning in the P systems area is the *total halting*, usually simply considered as *halting*:

Definition 4.20. An accessible configuration C is said to fulfill the *total halting* condition (H) if and only if no multiset of rules can be applied to C with respect to the derivation mode anymore, i.e.,

$$H\,(\Pi, \vartheta) \quad = \quad \{C' \mid C' \in Accessible\,(\Pi) \text{ and } Applicable\,(\Pi, C', \vartheta) = \emptyset\}\,.$$

The adult halting condition guarantees that we still can apply a multiset of rules to the underlying configuration, yet without changing it anymore:

Definition 4.21. An accessible configuration C is said to fulfill the *adult halting* condition (A) if and only if

$$\begin{aligned} A\,(\Pi, \vartheta) \quad = \quad & \{C' \mid C' \in Accessible\,(\Pi)\,,\ Applicable\,(\Pi, C', \vartheta) \neq \emptyset \text{ and} \\ & Apply\,(\Pi, C', R') = C' \text{ for every } R' \in Applicable\,(\Pi, C', \vartheta)\}\,. \end{aligned}$$

We should like to mention that we could also consider $A\,(\Pi, \vartheta) \cup H\,(\Pi, \vartheta)$ instead of $A\,(\Pi, \vartheta)$.

For introducing the notion of partial halting, we have to consider a partition of R into disjoint subsets R_1 to R_h as for the minimally parallel derivation mode. We then say that we are not halting only if there still is a multiset of rules R' from $Applicable\,(\Pi, C)$ with $R' \cap R_j \neq \emptyset$ for all j, $1 \leq j \leq h$:

Definition 4.22. An accessible configuration C is said to fulfill the *partial halting* condition (h) if and only if

$$h(\Pi, \vartheta) = \{C' \mid C' \in \mathit{Accessible}(\Pi) \text{ and there is no } R' \in \mathit{Applicable}(\Pi, C') \text{ with } R' \cap R_j \neq \emptyset \text{ for all } j,\ 1 \leq j \leq h\}.$$

4.2 Goal and result of a computation

The computations with a network of cells may have different goals, e.g., to generate (*gen*) a (vector of) non-negative integers in a specific output cell (membrane) or to accept (*acc*) a (vector of) non-negative integers placed in a specific input cell at the beginning of a computation. Moreover, the goal can also be to compute (*com*) an output from a given input or to output yes or no to decide (*dec*) a specific property of a given input.

The results not only can be taken as the number (N) of objects in a specified output cell, but, for example, also be taken modulo a terminal alphabet (T) or by subtracting a constant from the result $(-k)$.

Such different tasks of a network of cells may require additional parameters when specifying its functioning, e.g., we may have to specify the output/input cell(s) or the terminal alphabet.

We shall not go into the details of such definitions here, we just mention that the goal of the computations $\gamma \in \{gen, acc, com, dec\}$ and the way to extract the results ρ are two other parameters to be specified and clearly defined when defining the functioning of a network of cells or a membrane system.

4.3 Taxonomy of networks of cells and (tissue) P systems

For a particular variant of networks of cells or especially P systems/tissue P systems we have to specify the derivation mode, the halting condition as well as the procedure how to get the result of a computation, but also the specific kind of rules that are used, especially some complexity parameters.

For networks of cells, we shall use the notation

$$O_m C_n(\vartheta, \phi, \gamma, \rho)\,[\text{parameters for rules}]$$

to denote the family of sets of vectors obtained by networks of cells $\Pi = (V, w_1, w_2, \ldots, w_n, R)$ of degree n with $m = |V|$, as well as ϑ, ϕ, ρ indicating the derivation mode, the halting condition, and the way how to get results, respectively; the *parameters for rules* describe the specific features of the rules in R. If any of the parameters m and n is unbounded, we replace it by $*$.

For P systems, with the interaction between the cells in the rules of the corresponding network of cells allowing for a tree structure as underlying interaction graph, we shall use the notation

$$O_m P_n(\vartheta, \phi, \gamma, \rho)\,[\text{parameters for rules}].$$

Observe that usually the environment is not counted when specifying the number of membranes in P systems, but this usually hides that in many cases the environment takes an important role in the functioning of the system.

For tissue P systems, with the interaction between the cells in the rules of the corresponding network of cells allowing for a graph structure as underlying interaction graph, we shall use the notation

$$O_m tP_n\,(\vartheta, \phi, \gamma, \rho)\,[\text{parameters for rules}]\,.$$

As a special example, let us now consider *symport/antiport P systems.*

4.3.1 A specific example: P systems with symport/antiport rules

For definitions and results concerning P systems with symport/antiport rules, we refer to the original paper [13] as well as to the overview given in [16]. An *antiport rule* is a rule of the form $((x,i)\,(u,j) \to (x,j)\,(u,i))$ usually written as $(x, out; u, in)$, $xu \neq \lambda$, where j is the region outside the membrane i in the underlying graph structure. A *symport rule* is of the form $((x,i) \to (x,j))$ or $((u,j) \to (u,i))$.

The weight of the antiport rule $(x, out; u, in)$ is defined as $\max\{|x|\,, |u|\}$. Using only antiport rules with weight k induces the type of rules α usually written as $anti_k$. The weight of a symport rule (x, out) or (u, in) is defined as $|x|$ or $|u|$, respectively. Using only symport rules with weight k induces the type of rules α usually written as sym_k. If only antiport rules $(x, out; u, in)$ of weight ≤ 2 and with $|x| + |u| \leq 3$ as well as symport rules of weight 1 are used, we shall write $anti_{2'}$.

As is well known,

$$O_* P_2\,(max, H, gen, N)\,[anti_{2'}] = NRE.$$

Observe that we only need one membrane separating the environment and the skin region, but this means that two regions corresponding to two cells are involved.

4.3.2 A general result

For any network of cells using rules of type α, with a derivation mode ϑ, $\vartheta \in \{all_{aset}min, asyn, sequ\}$, and partial halting, we only get Parikh sets of matrix languages (regular sets of non-negative integers):

Theorem 1. For every $\vartheta \in \{all_{aset}min, asyn, sequ\}$,
$O_* C_*\,(\vartheta, h, gen, T)\,[\alpha] \subseteq PsMAT^{\lambda}$ and $O_* C_*\,(\vartheta, h, gen, N)\,[\alpha] \subseteq NREG$.

The proof follows the ideas of a similar result proved for a general variant of P systems with permitting contexts in [1] and therefore is omitted. We should like to mention that this result is still valid if we take the derivation mode $max_{set}min$ instead of $all_{aset}min$ (because when using partial halting we always have to take at least one rule from every set of rules), yet we do not know whether it also holds for the derivation modes min and/or $max_{rule}min$.

5 Conclusions

The main purpose of this paper is to elaborate a general framework for static P systems and tissue P systems, but there are many variants of membrane systems not yet covered by this general framework, especially dynamic changes of the number of cells cannot be handled with the current version. Yet we have already started to extend our approach to such dynamic variants like P systems with active membranes. Moreover, also spiking neural P systems require some efforts for being captured within this framework. Our approach aims at formalizing the main features of membrane systems in such a way that derivation modes and halting conditions can be defined in a clear and unambiguous way to avoid that different interpretations of notions and concepts in the P systems area yield incomparable results (as a special example consider the variants described for the minimally parallel derivation mode). Moreover, specifying the marking algorithm in a procedural way should allow for easier and unambiguous implementations. Considering variants of (tissue) P systems at such a high level of abstraction allows for establishing quite general results.

References

[1] A. Alhazov, R. Freund, M. Oswald, S. Verlan: Partial versus total halting in P systems. *Proc. Fifth Brainstorming Week on Membrane Computing*, Sevilla, 2007, *to appear*.

[2] O. Andrei, G. Ciobanu, D. Lucanu: A rewriting logic framework for operational semantics of membrane systems, *Theoretical Computer Science* **373**, 3 (2007), 163–181.

[3] F. Bernardini, M. Gheorghe, M. Margenstern, S. Verlan: Networks of Cells and Petri Nets. *Proc. Fifth Brainstorming Week on Membrane Computing*, Sevilla, 2007, *to appear*.

[4] G. Ciobanu, O. Andrei, D. Lucanu: Structural operational semantics of P systems. In: [9], 1–23.

[5] G. Ciobanu, L. Pan, Gh. Păun, M.J. Pérez-Jiménez: P systems with minimal parallelism, *accepted for TCS*.

[6] E. Csuhaj-Varjú, L. Kari, and G. Păun: Test tube distributed systems based on splicing. *Computers and AI* **15** (2–3) (1996), 211–232.

[7] E. Csuhaj-Varjú: Networks of Language Processors. *Current Trends in Theoretical Computer Science* (2001), 771–790.

[8] J. Dassow, Gh. Păun: On the power of membrane computing, *Journal of Universal Computer Science* **5** (2) (1999), 33–49.

[9] R. Freund, G. Lojka, M. Oswald, Gh. Păun (Eds.): *Pre-Proceedings of Sixth International Workshop on Membrane Computing (WMC6)*, Vienna, June 18-21, 2005.

[10] R. Freund, M. Oswald: P systems with partial halting, *accepted*, 2007.

[11] R. Freund, Gh. Păun, M.J. Pérez-Jiménez: Tissue-like P systems with channel states. *Theoretical Computer Science* **330** (2005), 101–116.

[12] J. Kleijn, M. Koutny, G. Rozenberg: Towards a Petri net semantics for membrane systems. In: [9], 439–460.

[13] A. Păun, Gh. Păun: The power of communication: P systems with symport/ antiport, *New Generation Computing* **20**, 3 (2002), 295–306.

[14] Gh. Păun: Computing with membranes, *J. of Computer and System Sciences* **61**, 1 (2000), 108–143, and TUCS Research Report 208 (1998) (`http://www.tucs.fi`).

[15] Gh. Păun: *Membrane Computing. An Introduction.* Springer-Verlag, Berlin, 2002.

[16] Y. Rogozhin, A. Alhazov, R. Freund: Computational power of symport/antiport: history, advances, and open problems. In: R. Freund, Gh. Păun, G. Rozenberg, A. Salomaa (Eds.): *Membrane Computing. 6th International Workshop WMC 2005*, Vienna, Austria, Lecture Notes in Computer Science **3850**, Springer-Verlag, 2006, 1–30.

[17] G. Rozenberg, A. Salomaa (Eds.): *Handbook of Formal Languages* (3 volumes), Springer-Verlag, Berlin, 1997.

[18] Gh. Păun, Y. Sakakibara, and T. Yokomori: P systems on graphs of restricted forms. *Publicationes Matimaticae* **60**, 2002.

[19] Gh. Păun and T. Yokomori: Membrane computing based on splicing. In: E. Winfree and D. K. Gifford (Eds.), *DNA Based Computers V*, volume 54 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 217–232. American Mathematical Society, 1999.

[20] The P Systems Web Page: `http://psystems.disco.unimib.it.`