Transition and Halting Modes for Tissue P Systems

Rudolf Freund

Faculty of Informatics, Vienna University of Technology
Favoritenstr. 9, 1040 Vienna, Austria
rudi@emcc.at

Summary. A variety of different transition modes for tissue P systems as well as several halting modes currently are used in the area of membrane computing. In this paper, the definitions of the most important transition modes and halting modes are explained based on networks of cells, a general model for tissue P systems. Moreover, some results for specific variants of tissue P systems working on multisets of objects are recalled.

1 Introduction

Membrane systems were introduced by Gheorghe Păun one decade ago as distributed parallel computing devices, based on inspiration from biochemistry, especially with respect to the structure and the functioning of a living cell, which is considered as a set of compartments enclosed by membranes containing objects and evolution rules. In the original model of membrane systems, the objects evolve in a hierarchical membrane structure (see [8], [16]); in tissue P systems (e.g., see [20], [21], and [11]), the cells communicate within an arbitrary graph topology. In the original model of membrane systems, the *maximally parallel transition mode* was used, yet later on also other new transition modes for P systems and tissue P systems have been introduced and investigated, for example, the *sequential* and the *asynchronous transition mode* as well as the *minimally parallel transition mode* (see [6]). In [12], a formal framework for (tissue) P systems capturing the formal features of these transition modes was developed, based on a general model of membrane systems as a collection of interacting cells containing multisets of objects (compare with the models of networks of cells as discussed in [5] and networks of language processors as considered in [7]). Continuing the formal approach started in [12], the *k-bounded minimally parallel transition mode* (see [13]) was introduced, where at most k rules can be taken from each of the sets of the partitioning of the set of rules used in the minimally parallel transition mode.

In most models of (tissue) P systems, a computation continues as long as still a (multiset of) rule(s) can be applied; the result of a computation then is taken at the end of a halting computation *(total halting)*. Recently, various other halting

conditions have been investigated; for example, when using *partial halting* (see [2], [3], [10]), a computation may only continue as long as from each set of a rule partitioning at least one rule can still be applied. The result of a computation may also be extracted at each step of a (halting or non-halting) computation, e.g., see [4].

The main parts of notions, definitions, and results presented in the following are taken from [12] and [13] as well as from [3] and [10]. For an introduction to the area of membrane computing we refer the interested reader to the monograph [17], the actual state of the art can be seen in the web [22].

2 Preliminaries

We recall some of the notions and the notations we use (for further details see [8] and [19]). 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. RE, REG ($RE(T)$, $REG(T)$) denote the families of recursively enumerable and regular languages (over the alphabet T), respectively; MAT^λ denotes the family of languages generated by context-free matrix grammars. 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; $[k..m]$ for $k \leq m$ denotes the set of natural numbers n with $k \leq n \leq m$. In the following, we will not distinguish between NRE, which coincides with $PsRE(\{a\})$, and $RE(\{a\})$.

Let V be a (finite) set, $V = \{a_1, ..., a_k\}$. A *finite multiset* M over V is a mapping $M : V \longrightarrow \mathbb{N}$, i.e., for each $a \in V$, $M(a)$ specifies the number of occurrences of a in M. The size of the multiset M is $|M| = \sum_{a \in V} M(a)$. A multiset M over V can also be represented by any string x that contains exactly $M(a_i)$ symbols a_i for all $1 \leq i \leq k$, e.g., by $a_1^{M(a_1)}...a_k^{M(a_k)}$. The set of all finite multisets over the set V is denoted by $\langle V, \mathbb{N} \rangle$. Throughout the rest of the paper, we will not distinguish between a multiset from $\langle V, \mathbb{N} \rangle$ and its representation by a string over V containing the corresponding number of each symbol. We may also consider mappings M of the form $M : V \longrightarrow \mathbb{N}_\infty$ where $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$, i.e., elements of M may have an infinite multiplicity; we shall call such multisets where $M(a_i) = \infty$ for at least one i, $1 \leq i \leq k$, *infinite multisets*. The set of all such multisets M over V with $M : V \longrightarrow \mathbb{N}_\infty$ is denoted by $\langle V, \mathbb{N}_\infty \rangle$.

3 Networks of Cells

In this section we consider membrane systems as a collection of interacting cells containing multisets of objects like in [5] and [12].

Definition 1. *A* network of cells – *we shall also use the notion* tissue P system – with checking sets, of degree $n \geq 1$, *is a construct*

$$\Pi = (n, V, w, R)$$

where

1. *n is the number of cells;*
2. *V is a* finite alphabet*;*
3. *$w = (w_1, \ldots, w_n)$ where $w_i \in \langle V, \mathbb{N}_\infty \rangle$, for all $1 \leq i \leq n$, is the* multiset initially associated to cell i *(in most of the cases, at most one cell, then being called* the environment*, will contain symbols occurring with infinite multiplicity);*
4. *R is a finite set of* rules *of the form*

$$(E : X \to Y)$$

where E is a recursive condition for configurations of Π (see definition below) as well as $X = (x_1, \ldots, x_n)$, $Y = (y_1, \ldots, y_n)$, with $x_i, y_i \in \langle V, \mathbb{N} \rangle$, $1 \leq i \leq n$, are vectors of multisets over V. We will also use the notation

$$(E : (x_1, 1) \ldots (x_n, n) \to (y_1, 1) \ldots (y_n, n))$$

for a rule $(E : X \to Y)$. If no conditions E are used, we use the simpler notations $X \to Y$ etc.

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 w_i. A *configuration* C of Π is an n-tuple of multisets over V $(u_1, \ldots, u_n)$; the *initial configuration* of Π, C_0, is described by w, i.e., $C_0 = w = (w_1, \ldots, w_n)$. Cells can interact with each other by means of the rules in R. An interaction rule

$$(E : (x_1, 1) \ldots (x_n, n) \to (y_1, 1) \ldots (y_n, n))$$

is applicable to a configuration C if and only if C fulfills condition E; its application means rewriting objects x_i from cells i into objects y_j in cells j, $1 \leq i, j \leq n$.

The set of all multisets of rules *applicable* to C is denoted by $Appl\,(\Pi, C)$ (a procedural algorithm how to obtain $Appl\,(\Pi, C)$ is described in [12]).

For the specific *transition modes* to be defined in the following, the selection of multisets of rules applicable to a configuration C has to be a specific subset of $Appl\,(\Pi, C)$; for the transition mode ϑ, the selection of multisets of rules applicable to a configuration C is denoted by $Appl\,(\Pi, C, \vartheta)$.

Definition 2. *For the* asynchronous *transition mode (asyn),*

$$Appl\,(\Pi, C, asyn) = Appl\,(\Pi, C)\,,$$

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

Definition 3. *For the* sequential *transition mode (sequ),*

$$Appl\,(\Pi, C, sequ) = \{R' \mid R' \in Appl\,(\Pi, C) \ \textit{and} \ |R'| = 1\}\,,$$

i.e., any multiset of rules $R' \in Appl\,(\Pi, C, sequ)$ *has size* 1.

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

Definition 4. *For the* maximally parallel *transition mode (max),*

$$\begin{aligned} Appl\,(\Pi, C, max) = \{R' \mid\ & R' \in Appl\,(\Pi, C) \ \textit{and there is} \\ & \textit{no } R'' \in Appl\,(\Pi, C) \ \textit{with } R'' \supsetneqq R'\}\,. \end{aligned}$$

For the *minimally parallel* transition 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. Usually, this partition of R may coincide with a specific assignment of the rules to the cells. 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 transition 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 [6]). For the basic variant as defined in the following, in each transition step we choose a multiset of rules R' from $Appl\,(\Pi, C, asyn)$ that cannot be extended to $R'' \in Appl\,(\Pi, C, asyn)$ 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 5. *For the* minimally parallel *transition mode (min),*

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

In [12], further restricting conditions on the four basic modes defined above, especially interesting for the minimally parallel transition mode, were considered. The following variant $all_{aset}min$ requires that from all applicable partition at least one rule has to be applied:

Definition 6. *For the* using all applicable sets minimally parallel *transition mode ($all_{aset}min$),*

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

We now consider a restricted variant of the minimally parallel transition mode allowing only a bounded number of at most k rules to be taken from each set R_j, $1 \leq j \leq h$, of the partitioning into a multiset of rules applicable in the minimally parallel transition mode.

Definition 7. *For the* k-restricted minimally parallel *transition mode* (min_k),

$$Appl\,(\Pi, C, min_k) = \{R' \mid R' \in Appl\,(\Pi, C, min) \text{ and } |R' \cap R_j| \leq k \text{ for all } j,\ 1 \leq j \leq h\}\,.$$

For all the transition 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 transition mode:

Definition 8. *Given a configuration* C *of* Π *and a transition mode* ϑ*, we may choose a multiset of rules* $R' \in Appl\,(\Pi, C, \vartheta)$ *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* $Apply\,(\Pi, C, R')$*, and we also write* $C \Longrightarrow_{(\Pi,\vartheta)} C'$*. The reflexive and transitive closure of the transition relation* $\Longrightarrow_{(\Pi,\vartheta)}$ *is denoted by* $\Longrightarrow^*_{(\Pi,\vartheta)}$*.*

Definition 9. *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* $Acc\,(\Pi)$*.*

Definition 10. *A derivation mode* ϑ *is said to be* deterministic *(det-*ϑ*) if* $|Appl\,(\Pi, C, \vartheta)| = 1$ *for any accessible configuration* C*.*

Definition 11. *A* computation *in a tissue P system* Π*,* $\Pi = (n, V, w, R)$*, starts with the initial configuration* $C_0 = w$ *and continues with transition steps according to the chosen transition mode* ϑ*.*

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

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

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 13. *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.,*

$$H(\Pi,\vartheta) = \{C' \mid C' \in Acc(\Pi) \text{ and } Appl(\Pi,C',\vartheta) = \emptyset\}.$$

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

Definition 14. *An accessible configuration C is said to fulfill the* adult halting *condition (*A*) if and only if*

$$A(\Pi,\vartheta) = \{C' \mid C' \in Acc(\Pi),\ Appl(\Pi,C',\vartheta) \neq \emptyset \text{ and } Apply(\Pi,C',R') = C' \text{ for every } R' \in Appl(\Pi,C',\vartheta)\}.$$

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

Definition 15. *An accessible configuration C is said to fulfill the* partial halting *condition (*h*) if and only if*

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

3.2 Goal and Result of a Computation

The computations with a tissue P system 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 tissue P system 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 tissue P system.

3.3 Taxonomy of Tissue P Systems

For a particular variant of networks of cells or tissue P systems we have to specify the transition 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 tissue P systems, we shall use the notation

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

to denote the family of sets of vectors obtained by tissue P systems $\Pi = (n, V, w, R)$ of degree n with $m = |V|$, as well as ϑ, ϕ, ρ indicating the transition 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 $*$.

If the communication structure in the tissue P system is a tree as in the original model of membrane systems, then we omit the t and use the notations $O_m C_n\,(\vartheta, \phi, \gamma, \rho)$ and $O_m C_n\,(\vartheta, \phi, \gamma, \rho)$.

4 Examples and Results

In this section, we give some examples how several well-known models of (tissue) P systems can be expressed within the general framework presented in the preceding section.

4.1 P Systems with Symport/Antiport Rules

For definitions and results concerning P systems with symport/antiport rules, we refer to the original paper [15] as well as to the overview given in [18]. 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 $anti_{2'}$. The following result is well known:

Theorem 1. $O_* tC_2\,(max, H, gen, N)\,[anti_{2'}] = NRE$.

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

4.2 Purely Catalytic P Systems

Already in the original paper of Gheorghe Păun (see [16]), membrane systems with catalytic rules were defined, but used together with other noncooperative rules. In [9] it was shown that only three catalysts are sufficient in one membrane, using only catalytic rules with the maximally parallel transition mode, to generate any recursively enumerable set of natural numbers.

A *noncooperative rule* is of the form $(I : (a,i) \rightarrow (y_1,1) \dots (y_n,n))$ where a is a single symbol and I denotes the condition that is always fulfilled. A *catalytic rule* is of the form $(I : (c,i)\,(a,i) \rightarrow (c,i)\,(y_1,1) \dots (y_n,n))$ where c is from a distinguished subset $C \subset V$ such that in all rules (noncooperative evolution rules, catalytic rules) of the whole system the y_i are from $(V-C)^*$ and the symbols a are from $(V-C)$. Imposing the restriction that the noncooperative rules and the catalytic rules in a tissue P system allow for finding a hierarchical tree structure of membranes such that symbols either stay in their membrane region or are sent out to the surrounding membrane region or sent into an inner membrane, then we get the classical catalytic P systems without priorities. Allowing regular sets checking for the non-appearance of specific symbols instead of I, we even get the original P systems with priorities. Catalytic P systems using only catalytic rules are called purely catalytic P systems. As we know from [9], only two (three) catalysts in one membrane are needed to obtain NRE with (purely) catalytic P systems without priorities working in the maximally parallel transition mode, i.e., we can write these results as follows:

Theorem 2. $NRE = O_*C_1\,(max, H, gen, -2)\,[cat_2]$
$= O_*C_1\,(max, H, gen, -3)\,[pcat_3]\,.$

If we now partition the rule set in a purely catalytic P system according to the catalysts present in each membrane, this partitioning replaces the use of the catalysts when working in the 1-restricted minimally parallel transition mode, because by definition from each of these sets then – if possible – exactly one rule (as with the use of the corresponding catalyst) is chosen: from the set of purely catalytic rules R we obtain the corresponding set of noncooperative rules R' as

$$R' = \{(a,i) \rightarrow (y_1,1) \dots (y_n,n) \mid (c,i)\,(a,i) \rightarrow (c,i)\,(y_1,1) \dots (y_n,n) \in R\}$$

as well as the corresponding partitioning of R' as

$$R'_{i,c} = \{(a,i) \rightarrow (y_1,1) \dots (y_n,n) \mid (c,i)\,(a,i) \rightarrow (c,i)\,(y_1,1) \dots (y_n,n) \in R\}\,.$$

Considering purely catalytic P systems in one membrane, we therefore infer the following quite astonishing result that when using the 1-restricted minimally parallel transition mode for a suitable partitioning of rules we only need noncooperative rules:

Theorem 3. $NRE = O_*C_1\,(min_1, H, gen, N)\,[noncoop]\,.$

When sing the asynchronous or the sequential transition mode, we only obtain regular sets:

Theorem 4. *For every* $\vartheta \in \{asyn, sequ\}$, $\phi \in \{H, h\}$, *and* $\gamma \in \{gen, acc\}$,

$$NREG = O_*tC_*\,(\vartheta, \phi, \gamma, N)\,[noncoop]\,.$$

4.3 Extended Spiking Neural P Systems

In extended spiking neural P systems (without delays, see [1]), the rules are applied in a sequential way in each neuron, but on the level of the whole system, the maximally parallel transition mode is applied (every neuron which may use a spiking rule has to spike, i.e., to apply a rule, see the original paper [14]). When partitioning the rule set according to the set of neurons, the application of the 1-restricted minimally parallel transition mode exactly models the original transition mode defined for spiking neural P systems.

An *extended spiking neural P system* (of degree $m \geq 1$) (in the following we shall simply speak of an *ESNP system*) is a construct $\Pi = (m, S, R)$ where

- m is the number of *neurons*; the neurons are uniquely identified by a number between 1 and m;
- S describes the *initial configuration* by assigning an initial value (of spikes) to each neuron;
- R is a finite set of *rules* of the form $\left(i, E/a^k \to P\right)$ such that $i \in [1..m]$ (specifying that this rule is assigned to neuron i), $E \subseteq REG\,(\{a\})$ is the *checking set* (the current number of spikes in the neuron has to be from E if this rule shall be executed), $k \in \mathbb{N}$ is the "number of spikes" (the energy) consumed by this rule, and P is a (possibly empty) set of *productions* of the form (l, a^w) where $l \in [1..m]$ (thus specifying the target neuron), $w \in \mathbb{N}$ is the *weight* of the energy sent along the axon from neuron i to neuron l.

A *configuration* of the ESNP system is described by specifying the actual number of spikes in every neuron. A *transition* from one configuration to another one is executed as follows: for each neuron i, we non-deterministically choose a rule $\left(i, E/a^k \to P\right)$ that can be applied, i.e., if the current value of spikes in neuron i is in E, neuron i "spikes", i.e., for every production (l, w) occurring in the set P we send w spikes along the axon from neuron i to neuron l. A *computation* is a sequence of configurations starting with the initial configuration given by S. An ESNP system can be used to generate sets from NRE (we do not distinguish between NRE and $RE\,(\{a\})$) taking the contents, i.e., the number of spikes, of a specific neuron called *output neuron* in halting computations.

We now consider the ESNP system $\Pi = (m, S, R)$ as a tissue P system $\Pi' = (m, \{a\}, S, R')$ working in the 1-restricted minimally parallel transition mode, with

$$R' = \big\{ \big(E : \big(a^k, i\big) \to (a^{w_1}, l_1) \dots (a^{w_n}, l_n)\big) \mid \big(i, E/a^k \to (l_1, a^{w_1}) \dots (l_n, a^{w_n})\big) \in R \big\}$$

and the partitioning R'_i, $1 \leq i \leq m$, of the rule set R' according to the set of neurons, i.e.,

$$R'_i = \big\{ \big(E : \big(a^k, i\big) \to (a^{w_1}, l_1) \dots (a^{w_n}, l_n)\big) \mid \big(E : \big(a^k, i\big) \to (a^{w_1}, l_1) \dots (a^{w_n}, l_n)\big) \in R' \big\} .$$

The 1-restricted minimally parallel transition mode chooses one rule – if possible – from every set R_i and then applies such a multiset of rules in parallel, which directly corresponds to applying one spiking rule in every neuron where a rule can be applied. Hence, it is easy to see that Π' and Π generate the same set from $RE\{a\}$ if in both systems we take the same cell/neuron for extracting the output. Due to the results valid for ESNP systems, see [1], we obtain the following result:

Theorem 5. $NRE = O_1tC_3\,(min_1, H, gen, N)\,[ESNP]$.

4.4 A General Result

For any tissue P system using rules of type α, with a transition 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), provided the checking set for each rule can be simulated by checking the (independent) applicability of a finite set of rules (fixed for each rule):

Theorem 6. *For every* $\vartheta \in \{all_{aset}min, asyn, sequ\}$,

$$O_*tC_*\,(\vartheta, h, gen, T)\,[\alpha] \subseteq PsMAT \text{ and}$$
$$O_*tC_*\,(\vartheta, h, gen, N)\,[\alpha] \subseteq NREG.$$

The proof follows the ideas of a similar result proved for a general variant of P systems with permitting contexts in [3] and therefore is omitted. We do not know whether a similar result also holds true for the transition mode min itself instead of $all_{aset}min$.

5 Conclusions

In the general framework considered in this paper, many variants of static tissue P systems (and P systems as well) can be represented. Although during the last decade, a great variety of such systems working in different transition mode has been considered, many specific models of (tissue) P systems still wait for being considered with other transition modes, for example, with the k-restricted minimally parallel transition mode. Moreover, different variants of halting, especially partial halting, should be considered for a lot more models of (tissue) P systems in the future.

Acknowledgements

I am very grateful to the coauthors of those papers from which most of the definitions and results have been taken for this paper, especially to Artiom Alhazov, Marion Oswald, and Sergey Verlan, yet most of all to Gheorghe Păun, whose great ideas have inspired myself to develop many new variants of P systems.

References

1. A. Alhazov, R. Freund, M. Oswald, M. Slavkovik: Extended spiking neural P systems generating strings and vectors of non-negative integers, in: H.J. Hoogeboom, Gh. Paun, G. Rozenberg (Eds.): *Pre-proceedings of Membrane Computing, International Workshop, WMC7*, Leiden, The Netherlands, 2006, 88–101.
2. A. Alhazov, R. Freund, M. Oswald, S. Verlan: Partial versus total halting in P systems, in: M.A. Gutiérrez-Naranjo, Gh. Păun, M.J. Pérez-Jiménez (Eds.): *Cellular Computing (Complexity Aspects)*, ESF PESC Exploratory Workshop, Fénix Editorial, Sevilla, 2005, 1–20.
3. A. Alhazov, R. Freund, M. Oswald, S. Verlan: Partial halting in P systems using membrane rules with permitting contexts, in: J.O. Durand-Lose, M. Margenstern (Eds.): *Proc. of MCU 2007*, Orléans, France, LNCS **4664**, Springer, 2007, 110–121.
4. M. Beyreder, R. Freund: (Tissue) P systems using noncooperative rules without halting conditions, in: P. Frisco et al. (Eds.): *Pre-Proc. Ninth Workshop on Membrane Computing (WMC9),* Edinburgh, 2008, 85–94.
5. F. Bernardini, M. Gheorghe, M. Margenstern, S. Verlan: Networks of Cells and Petri Nets, in: M. A. Gutiérrez-Naranjo, Gh. Păun, A. Romero-Jiménez, A. Riscos-Núñez (Eds.): Proc. Fifth Brainstorming Week on Membrane Computing, Sevilla, 2007, 33–62.
6. G. Ciobanu, L. Pan, Gh. Păun, M.J. Pérez-Jiménez: P systems with minimal parallelism, *Theoretical Computer Science* **378** (1) (2007), 117–130.
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, L. Kari, M. Oswald, P. Sosík, Computationally universal P systems without priorities: two catalysts are sufficient, *Theoretical Computer Science* **330** (2005), 251–266.
10. R. Freund, M. Oswald: Partial halting in P systems. *Intern. J. Foundations of Computer Sci.* **18** (2007), 1215–1225.
11. R. Freund, Gh. Păun, M.J. Pérez-Jiménez: Tissue-like P systems with channel states. *Theoretical Computer Science* **330** (2005), 101–116.
12. R. Freund, S. Verlan, A formal framework for P systems, in: G. Eleftherakis, P. Kefalas, Gh. Paun (Eds.): *Pre-proceedings of Membrane Computing, International Workshop – WMC8*, Thessaloniki, Greece, 2007, 317–330.
13. R. Freund, S. Verlan, (Tissue) P systems working in the k-restricted minimally parallel derivation mode, in: E. Csuhaj-Varjú, R. Freund, M. Oswald, K. Salomaa (Eds.): *Proceedings of the International Workshop on Computing with Biomolecules*, Österreichische Computer Gesellschaft, 2008, 43–52.

14. M. Ionescu, Gh. Păun, T. Yokomori: Spiking neural P systems, *Fundamenta Informaticae* **71**, 2–3 (2006), 279–308.
15. A. Păun, Gh. Păun: The power of communication: P systems with symport/ antiport, *New Generation Computing* **20** (3) (2002), 295–306.
16. 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`).
17. Gh. Păun: *Membrane Computing. An Introduction.* Springer-Verlag, Berlin, 2002.
18. 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.
19. G. Rozenberg, A. Salomaa (Eds.): *Handbook of Formal Languages* (3 volumes), Springer-Verlag, Berlin, 1997.
20. Gh. Păun, Y. Sakakibara, and T. Yokomori: P systems on graphs of restricted forms. *Publicationes Matimaticae* **60**, 2002.
21. 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.
22. The P Systems Web Page: `http://ppage.psystems.eu.`