Hybrid Transition Modes in (Tissue) P Systems

Rudolf Freund and Marian Kogler

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

Summary. In addition to the maximally parallel transition mode used from the beginning in the area of membrane computing, many other transition modes for (tissue) P systems have been investigated since then. In this paper we consider (tissue) P systems with hybrid transition modes where each set of a partitioning of the whole set of rules may work in a different transition mode in a first level and all partitions of rules work together at a (second) level of the whole system on the current configuration in a maximally parallel way. With all partitions of noncooperative rules working in the maximally parallel mode, we obtain a characterization of Parikh sets of $ET0L$-languages, whereas with hybrid systems with either the partitions working in the maximally parallel as well as in the $=1$-mode or with all partitions working in the $=1$-mode we can simulate catalytic or purely catalytic P systems, respectively, thus obtaining computational completeness.

1 Introduction

In the original model of P systems introduced as membrane systems by Gh. Păun (see [6], [12]), the objects evolve in a hierarchical membrane structure; in tissue P systems, for example considered by Gh. Păun, T. Yokomori, and Y. Sakakibara in [15] and by R. Freund, Gh. Păun, and M.J. Pérez-Jiménez in [8], the cells communicate within an arbitrary graph topology. The maximally parallel transition mode was not only used in the original model of membrane systems, but then also in many variants of P systems and tissue P systems investigated during the last decade. Rather recently several 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 [3]) and the k-bounded minimally parallel transition mode (see [10]). In [9], 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 [1] and networks of language processors as considered in [4]). In this paper we consider partitionings of the rule set with each partition being equipped with its own transition mode – which may not only be the transition modes usually considered in the area of P systems as the maximally parallel mode, but also modes well known from the area of grammar systems (e.g., see [5]) as the $= k$, $\leq k$, and the $\geq k$ modes for $k \geq 1$. A multiset of rules to be applied to a given configuration is composed from a multiset of rules from each partition working in the corresponding transition mode on a suitable partitioning of the objects in the underlying configuration.

The rest of this paper is organized as follows: In the second section, well-known definitions and notions are recalled. In the next section, we explain our general model of tissue P systems with hybrid transition modes and give some illustrative examples in the succeeding section. A characterization of the Parikh sets of $ET0L$-languages by tissue P systems with all partitions working in the maximally parallel transition mode is shown in the fourth section. In the fifth section, we establish some results on computational completeness by showing how catalytic P systems and purely catalytic P systems can be simulated by tissue P systems where one partition works in the maximally parallel mode and all the others in the $= 1$-mode and by tissue P systems where all partitions work in the $= 1$-mode, respectively. A short summary concludes the paper.

2 Preliminaries

We recall some of the notions and the notations we use (see [14] for elements of formal language theory) as in [10].

Let V be a (finite) alphabet; then V^* is the set of all strings 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. For any family of string languages F, PsF denotes the family of Parikh sets of languages from F. 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. 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.

An ET0L system is a construct $G = (V, T, w, P_1, \ldots, P_m)$, $m \geq 1$, where V is an alphabet, $T \subseteq V$ is the terminal alphabet, $w \in V^*$ is the *axiom*, and P_i, $1 \leq i \leq m$, are finite sets of rules (*tables*) of noncooperative rules over V of the form $a \rightarrow x$. In a derivation step, all the symbols present in the current sentential form are rewritten using one table. The language generated by G, denoted by $L(G)$, consists of all the strings over T which can be generated in this way when starting from w. An ET0L system with only one table is called an E0L system. By $E0L$ and $ET0L$ we denote the families of languages generated by E0L systems and ET0L systems, respectively. It is known from [14] that $CF \subset E0L \subset ET0L \subset CS$, with CF being the family of context-free languages and CS being the family of context-sensitive languages. The corresponding families of sets of (vectors of) non-negative integers are denoted by XCF, $XE0L$, $XET0L$, and XCS, respectively, with $X \in \{N, Ps\}$.

A *register machine* is a construct $M = (n, B, l_0, l_h, I)$, where n is the number of registers, B is a set of instruction labels, l_0 is the start label, l_h is the halt label (assigned to `HALT` only), and I is a set of instructions of the following forms:

- $l_i : (\mathtt{ADD}(r), l_j, l_k)$ add 1 to register r, and then go to one of the instructions labeled by l_j and l_k, non-deterministically chosen;
- $l_i : (\mathtt{SUB}(r), l_j, l_k)$ if register r is non-empty (non-zero), then subtract 1 from it and go to the instruction labeled by l_j, otherwise go to the instruction labeled by l_k;
- $l_h : \mathtt{HALT}$ the halt instruction.

A register machine M generates a set $N(M)$ of natural numbers in the following way: start with the instruction labeled by l_0, with all registers being empty, and proceed to apply instructions as indicated by the labels and by the contents of the registers. If we reach the `HALT` instruction, then the number stored at that time in register 1 is taken into $N(M)$. It is known (e.g., see [11]) that in this way we can compute all recursively enumerable sets of natural numbers even with only three registers, where the first one is never decremented.

3 Networks of Cells

In this section we consider membrane systems as a collection of interacting cells containing multisets of objects like in [1] and [9]. For an introduction to the area of membrane computing we refer the interested reader to the monograph [13], the actual state of the art can be seen in the web [17].

Definition 1. *A* network of cells of degree $n \geq 1$ *is a construct*

$$\Pi = (n, V, w, i_0, R) \quad \text{where}$$

1. n *is the* number of cells*;*
2. V *is a (finite)* alphabet*;*
3. $w = (w_1, \dots, w_n)$ *where* $w_i \in \langle V, \mathbb{N} \rangle$*, for all* $1 \leq i \leq n$*, is the* multiset initially associated to cell i*;*
4. i_0*,* $1 \leq i_0 \leq n$*, is the output cell;*

5. *R is a finite set of* rules *of the form $X \to Y$ where $X = (x_1, \ldots, x_n)$, $Y = (y_1, \ldots, y_n)$, 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*

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

for a rule $X \to Y$.

A network of cells consists of n cells, numbered from 1 to n, that contain 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. The application of a rule

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

means rewriting objects x_i from cells i into objects y_j in cells j, $1 \leq i, j \leq n$. A rule is called *noncooperative* if it is of the form $(a, i) \to (y_1, 1) \ldots (y_n, n)$ with $a \in V$.

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 [9]).

We now consider a partition of R into disjoint subsets R_1 to R_h, $h \geq 1$. Usually, this partition of R may coincide with a specific assignment of the rules to the cells, yet in this paper we do not restrict ourselves to such a constraint, but allow the rule sets R_1 to R_h to be working on arbitrary cells. For any multiset of rules R' containing rules from a set of rules R, we define $\|R'\|$ to be the number of rules in R'.

For the specific *transition modes* used for the subsets of rules R_j to be defined in the following, we consider the subsystems

$$\Pi_j = (n, V, w, i_0, R_j)\,.$$

The selection of multisets of rules from R_j, $1 \leq j \leq h$, applicable to a configuration C has to be a specific subset of $Appl\,(\Pi_j, C)$; for the transition mode ϑ, the selection of multisets of rules applicable to a configuration C is denoted by $Appl\,(\Pi_j, C, \vartheta)$. In contrast to the transition modes usually considered in the area of P systems as the asynchronous and the sequential mode, we also define some more general variants well known from the area of grammar systems (e.g., see [5]) as the derivation modes $= k$, $\geq k$, $\leq k$ for $k \geq 1$.

Definition 2. *For the transition mode (Δk) with $\Delta \in \{=, \leq, \geq\}$,*

$$Appl\,(\Pi_j, C, \Delta k) = \{R' \mid R' \in Appl\,(\Pi_j, C) \text{ and } \|R'\|\, \Delta k\}\,.$$

The *asynchronous* transition mode ($asyn$) with

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

is the special case of the transition mode Δk with Δk being equal to ≥ 1, i.e., in fact there are no particular restrictions on the multisets of rules applicable to C.

The *sequential* transition mode (*sequ*) with

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

is the special case of the transition mode Δk with Δk being equal to $= 1$, i.e., every multiset of rules $R' \in Appl\,(\Pi_j, C, sequ)$ has size 1.

The 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 3. *For the* maximally parallel *transition mode (max),*

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

Based on these transition modes for the partitions of rules R_j, we now are able to define a *network of cells with hybrid transition modes* as follows:

Definition 4. *A* network of cells with hybrid transition modes of degree $n \geq 1$*, in the following also called* tissue P system (with hybrid transition modes) of degree $n \geq 1$*, is a construct*

$$\Pi = (n, V, w, i_0, R, (R_1, \alpha_1), \ldots, (R_h, \alpha_h)) \quad \text{where}$$

1. (n, V, w, i_0, R) *is a network of cells of degree* n*;*
2. $R_1, \ldots, R_h$ *is a partition of* R *into disjoint subsets* R_1 *to* R_h *and the* α_j*,* $1 \leq j \leq h$*, are the transition modes assigned to the corresponding partitions of rules* R_j*.*

Based on the transition modes of the partitions R_j, we now can define how to obtain a next configuration from a given one in the whole system Π by applying in a maximally parallel way an applicable multiset of rules consisting of multisets of rules from the R_j each of those applied in the respective transition mode:

Definition 5. *Given a configuration* C *of* Π*, we non-deterministically choose a partition* R_{j_1} *and try to apply it; if this is not possible, we just continue with non-deterministically choosing another partition* R_{j_2}*; if we are able to apply* R_{j_1} *in the corresponding transition mode* α_{j_1} *with using a multiset of rules* R'_{j_1}*, we mark the objects affected by doing that and continue with non-deterministically choosing another partition* R_{j_2} *then being to be applied to a configuration not containing the objects marked for being used with the rules from* R'_{j_1}*. We continue with the same algorithm as for* R_{j_1} *eventually marking objects to be used with a multiset of rules* R'_{j_2}*, etc. In sum, we obtain a multiset of rules* R' *to be applied to* C *as the union of the multisets of rules* R'_{j_m} *constructed by the algorithm described* *above. The result of the* transition step *from the configuration C with applying R' is the configuration $Apply\,(\Pi, C, R')$, and we also write $C \Longrightarrow_{\Pi} C'$. The reflexive and transitive closure of the transition relation $\Longrightarrow_{\Pi}$ is denoted by $\Longrightarrow_{\Pi}^{*}$; if n transition steps take place, we write $\Longrightarrow_{\Pi}^{n}$ for $n \geq 0$.*

Definition 6. *A* computation *in a network of cells with hybrid transition modes Π starts with the initial configuration $C_0 = w$ and continues with transition steps as defined above. It is called* successful *if we reach a configuration C to which no partition R_j can be applied with respect to the transition mode α_j anymore (we also say that the computation* halts*).*

Definition 7. *As the* results *of halting computations we take the Parikh vectors or numbers of objects in the specified output cell i_0. The set of results of all computations then is denoted by $X\,(\Pi)$ with $X \in \{Ps, N\}$.*

We shall use the notation $XO_m h_h t P_n\,(\vartheta)\,[parameters\ for\ rules]$ with $X \in \{Ps, N\}$ to denote the family of sets of Parikh vectors (Ps) and natural numbers (N), respectively, generated by tissue P systems Π of the form

$$(n', V, w, i_0, R, (R_1, \alpha_1), \ldots, (R_{h'}, \alpha_{h'}))$$

with $n' \leq n$, $|V| \leq m$, $h' \leq h$, and $\cup_{j=1}^{h} \{\alpha_j\} \subseteq \vartheta$ (ϑ contains the allowed transition modes); the parameters for rules *describe the specific features of the rules in R. If any of the parameters n, m, and h is unbounded, we replace it by $*$.*

4 Examples

As a first example, we construct a tissue P system with one cell initially containing two symbols a and two sets of rules each of them containing one rule affecting the symbol a using eventually different transition modes:

Example 1. Let

$$\Pi = (1, \{a\}, aa, 1, P_1 \cup P_2, (P_1, \alpha_1), (P_2, \alpha_2))$$

where $P_1 = \{a \to b\}$ and $P_2 = \{a \to c\}$. We now consider the results of computations in this tissue P system with different transition modes α_1 and α_2:

- α_1 and α_2 both are $= 1$: both the rule in P_1 and the rule in P_2 are applied exactly once, no matter which partition we choose first to be applied, i.e., $aa \Longrightarrow_{\Pi} bc$; hence, the result is bc.
- α_1 and α_2 both are max: recall that the transition modes of the rule sets do not take into account the rules in other rule sets, so both P_i try to apply their own rule twice. This conflict is solved in a non-deterministic way, i.e., $aa \Longrightarrow_{\Pi_1} bb$ or $aa \Longrightarrow_{\Pi_2} cc$; hence, the results are bb, cc.

- α_1 and α_2 both are ≥ 1: the rules in P_i are applied either once or twice. If the rule from each set is only applied once, we have a similar situation as before when using the transition mode $= 1$. If one or both sets attempt to apply their own rule twice, a conflict arises which is solved in a non-deterministic way. Thus, the result set is the union of the result sets considered in the cases $= 1$ and max, i.e., $\{bc, bb, cc\}$.
- α_1 is $= 1$, α_2 is ≥ 1: as before, yet we do not have to consider the case that the rule in P_1 is applied twice. Therefore, the result set is $\{bc, cc\}$.
- α_1 is $= 1$, α_2 is max: the conflict is solved by non-deterministically choosing to execute the rule in P_2 in a maximally parallel way thus consuming all symbols a before trying to execute the rule in P_1 (which then fails, as no symbol a is left) or else to execute P_1 before P_2 (resulting in one a transformed to b and one a transformed to c). This yields the same result set as in the case before ($\{bc, cc\}$).
- α_1 is ≥ 1, α_2 is max: P_1 and P_2 conflict either with respect to one symbol (if the rule in the partition chosen first is applied only once) or over both symbols (if it is applied twice). If the conflict arises with respect to one symbol, the conflict resolution yields $\{bc, cc\}$; otherwise, as in the case when α_1 and α_2 both are max, the results are bb, cc. The set of all possible computation results thus is the union of both cases, i.e., $\{bc, bb, cc\}$.

Usually, with only taking results from halting computations and using the maximally parallel transition mode without partitioning the rule set R, with non-cooperative rules it is not possible to generate sets like $\left\{a^{2^n} \mid n \geq 0\right\}$ (compare with the results established in [2], where the variant of unconditional halting was used instead, i.e., the results were taken in every computation step). As the following example shows, such sets can easily obtained with specific partitions of non-cooperative rules all of them working in the maximally parallel transition mode:

Example 2. Consider the tissue P system (of degree 1)

$$\Pi = (1, \{a, b\}, b, 1, P_1 \cup P_2, (P_1, max), (P_2, max))$$

with $P_1 = \{b \to bb\}$ and $P_2 = \{b \to a\}$. As elaborated in the previous example, we can either apply $b \to bb$ OR $b \to a$ in a maximally parallel way, but not mix both rules. Hence, as long as we apply P_1 in the maximally parallel mode, in each transition step we double the number of objects b. As soon as we choose to apply P_2 in the maximally parallel mode, the computation comes to an end yielding a^{2^n} for some $n \geq 0$, i.e., $b \Longrightarrow_\Pi^n b^{2^n} \Longrightarrow_\Pi a^{2^n}$, hence, $X(\Pi) = \left\{a^{2^n} \mid n \geq 0\right\}$ with $X \in \{Ps, N\}$.

5 Characterization of $ET0L$

In this section we show that tissue P systems with all partitions (of noncooperative rules) working in the maximally parallel transition mode exactly yield the (Parikh sets of) $ET0L$-languages.

Theorem 1. *$PsET0L = PsO_*h_*tP_n\left(\{max\}\right)[noncoop]$ for all $n \geq 1$.*

Proof. We first show $PsET0L \supseteq PsO_*h_*tP_*\left(\{max\}\right)$ [noncoop]. Let

$$\Pi = \left(n, V, w, i_0, R, \left(R_1, max\right), \ldots, \left(R_h, max\right)\right)$$

be a tissue P system with hybrid transition modes with all partitions working in the max-mode. We first observe that an object a from V in the cell m, $1 \leq m \leq n$, can be represented as a new symbol (a, m). Hence, in the ET0L-system

$$G = \left(V', T, w', P_1, \ldots, P_d, P_f\right)$$

simulating Π, we take $T = V$ and $V' = V'' \cup V \cup \{\#\}$ with

$$V'' = \{(a, m) \mid a \in V, 1 \leq m \leq n\}.$$

In the axiom w', every symbol a in cell m is represented as the new symbol (a, m). Observe that a noncooperative rule

$$(a, i) \rightarrow (y_1, 1) \ldots (y_n, n)$$

can also be written as

$$(a, i) \rightarrow (y_{1,1}, 1) \ldots (y_{1,d_1}, 1) \ldots (y_{n,1}, 1) \ldots (y_{n,d_n}, n)$$

where all $y_{i,j}$ are objects from V and in that way can just be considered as a pure context-free rule over V''.

For every sequence of partitions $l = \langle R'_1, \ldots, R'_h \rangle$ such that $\{R'_1, \ldots, R'_h\} = \{R_1, \ldots, R_h\}$, we now construct a table P_l for G as follows:

$P_l := \left\{x \rightarrow x \mid x \in V', x \neq y \text{ for all rules } y \rightarrow v \text{ in } \cup_{i=1}^{h} R_i\right\}$;
for $i = 1$ to h do
  begin
  $R_i^{''} := \{x \rightarrow w \mid x \rightarrow w \in R'_i \text{ and } x \neq y \text{ for all rules } y \rightarrow v \text{ in } P_l\}$;
  $P_l := P_l \cup R_i^{''}$
  end

As all partitions work in the max-mode, a partition applied first consumes all objects for which it has suitable rules. Finally, to fulfill the completeness condition for symbols usually required in the area of Lindenmayer systems, we have added unit rules $a \rightarrow a$ for all objects not affected by the rule sets $R_1, \ldots, R_h$. In that way, one transition step in Π with using a multiset of rules marking the objects in the underlying configuration according to the sequence of partitions $\langle R_1', \dots, R_h' \rangle$ exactly corresponds with an application of the table P_l in G. To extract the terminal configurations, we have to guarantee that no rule from $\cup_{i=1}^{h} R_i$ can be applied anymore (they are projected on the trap symbol $\#$) and project the symbols (a, i_0) from the output membrane to the terminal symbols a, which is accomplished by the final table

$$\begin{aligned} P_f := & \left\{ x \to \# \mid x \in V'' \text{ for some rule } x \to v \text{ in } \cup_{i=1}^{h} R_i \right\} \cup \{\# \to \#\} \\ \cup & \ \{(a,i) \to \lambda \mid a \in V, i \neq i_0 \text{ and there is no rule } (a,i) \to v \text{ in } R_i\} \\ \cup & \ \{(a,i_0) \to a \mid a \in V \text{ and there is no rule } (a,i_0) \to v \text{ in } R_{i_0}\}. \end{aligned}$$

We now show the inclusion$PsET0L \subseteq PsO_*h_*tP_1 (\{max\}) [\text{noncoop}]$.

Let $G = (V, T, w, P_1, \dots, P_n)$ be an ET0L-system. Then we construct the equivalent tissue P system with only one cell and $n + 2$ partitions all of them working in the maximally parallel mode

$$\Pi = (1, V \cup T' \cup \{\#\}, h(w), 1, R, (R_1, max), \dots, (R_{n+2}, max))$$

as follows:

The renaming homomorphism $h : V \to (V - T) \cup T'$ is defined by $h(a) = a$ for $a \in V - T$ and $h(a) = a'$ for $a \in T$. Then we simply define $R_i = h(P_i)$ for $1 \leq i \leq n$, i.e., in all rules we replace every terminal symbol a from T by its primed version a'. If G has produced a terminal multiset, then Π should stop with yielding the same result, which is accomplished by applying the partition

$$R_{n+1} = \{a' \to a \mid a \in T\} \cup \{x \to \# \mid x \in V - T\};$$

if the terminating rule set R_{n+1} is applied while objects from $V - T$ are still present, trap symbols $\#$ are generated, which causes a non-terminating computation in Π because of the partition $R_{n+2} = \{\# \to \#\}$. These observations conclude the proof. □

6 Simulation of (Purely) Catalytic P Systems and Computational Completeness

Membrane systems with catalytic rules were already defined in the original paper of Gh. Păun (see [12]), but used together with noncooperative rules. In the notations of this paper, a *noncooperative rule* is of the form $(a, i) \to (y_1, 1) \dots (y_n, n)$, and a *catalytic rule* is of the form $(c, i)(a, i) \to (c, i)(y_1, 1) \dots (y_n, n)$ where c is from a distinguished subset $C \subset V$ such that in all rules – noncooperative rules *(noncoop)* and catalytic rules *(cat)* of the whole system – the y_i are from $(V - C)^*$ and the symbols a are from $(V - C)$.

A *catalytic tissue P system* can be written as a tissue P system with hybrid transition modes for rule partitions

$$\Pi = (n, V, C, w, i_0, R, (R, max))$$

where the single rule set R works in the maximally parallel transition mode and the rules are noncooperative rules and catalytic rules. If all rules in R are catalytic ones, such a system is called *purely catalytic*. We have to point out that in the following we shall assume that each catalyst can appear only once in the whole system; as catalysts cannot move from one cell to another one, this assumption is no restriction of generality. Moreover, we recall the fact that in the catalytic tissue P systems as defined above we allow arbitrary connections between cells, whereas in the original variant of catalytic P systems, the connection graph is restricted to a tree. As a technical detail we mention that catalysts appearing in the output cell are not taken into account when extracting the results of a computation.

By $XO_mC_ktP_n\,[cat]$ ($XO_mC_ktP_n\,[pcat]$) with $X \in \{Ps, N\}$ we denote the family of sets of Parikh vectors (Ps) and natural numbers (N), respectively, generated by (purely) catalytic tissue P systems of the form $(n', V, C, w, i_0, R, (R, max))$ with $n' \leq n$, $|V| \leq m$, and $|C| \leq k$. If any of the parameters n, m, and k is unbounded, we replace it by $*$.

We now show that catalytic tissue P systems can be simulated by tissue P systems with hybrid transition modes for rule partitions using the maximally parallel transition mode for one partition and the $= 1$-mode for all other partitions of rules:

Theorem 2. *$XO_mC_ktP_n\,[cat] \subseteq XO_mh_{k+1}tP_n\,(\{max, =1\})\,[noncoop]$ for $X \in \{Ps, N\}$ and all natural numbers m, k, and n.*

Proof. Let $\Pi = (n, V, C, w, i_0, R, (R, max))$ be a catalytic tissue P system with n cells. Then we construct an equivalent tissue P system with hybrid transition modes for rule partitions Π' as follows:

$$\Pi' = (n, V, w, i_0, R, (R_1, =1), \ldots, (R_k, =1), (R_{k+1}, max))$$

where, for $C = \{c_j \mid 1 \leq j \leq k\}$,

$$R_j = \{(a, i) \rightarrow (y_1, 1) \ldots (y_n, n) \mid (c_j, i)\,(a, i) \rightarrow (c_j, i)\,(y_1, 1) \ldots (y_n, n) \in R\}$$

for $1 \leq j \leq k$ and $R_{k+1} = R - \cup_{j=1}^{k} R_j$. For each catalyst c_j, the catalytic rules involving c_j form the partition R_j, from which at most one rule can be taken in any transition step, i.e., the R_j, $1 \leq j \leq k$, are combined with the $= 1$-mode, and the remaining noncooperative rules from R are collected in R_{k+1} and used in the max-mode. The equivalence of the systems Π' and Π immediately follows from the definition of the respective transition modes and the resulting transitions in these systems. □

From the proof of the preceding theorem, we immediately infer the following result for purely catalytic tissue P systems:

Corollary 1. *For $X \in \{Ps, N\}$ and all natural numbers m, k, and n,*

$$XO_mC_ktP_n\,[pcat] \subseteq XO_mh_ktP_n\,(\{=1\})\,[noncoop]\,.$$

In [7] it was shown that only three catalysts are sufficient in one cell, using only catalytic rules with the maximally parallel transition mode, to generate any recursively enumerable set of natural numbers. Hence, by showing that (tissue) P systems with purely catalytic rules working in the maximally parallel transition mode can be considered as tissue P systems with partitions of corresponding noncooperative rules working in the $=1$-mode when partitioning the rule set for the single cell with respect to the catalysts, we obtain the interesting result that in this case we get a characterization of the recursively enumerable sets of natural numbers by using only noncooperative rules (in fact, this partitioning replaces the use of the catalysts). In sum, from Theorem 2 and Corollary 1 and the results from [7] we obtain the following result showing computational completeness for tissue P systems with hybrid transition modes for rule partitions:

Theorem 3. $NRE = NO_*h_3tP_1\,(\{=1\})\,[noncoop]$
$NO_*h_3tP_1\,(\{max, =1\})\,[noncoop]\,.$

We mention that the $=1$ mode in any case can be replaced by the ≤ 1-mode which immediately follows from the definition of the respective transition modes. Moreover, having the partitions working in the $=1$-mode on the first level and using maximal parallelism on the second level of the whole system corresponds with the min_1 transition mode as introduced in [10] - this min_1 transition mode forces to take exactly one rule or zero rules from each partition into an applicable multiset of rules in such a way that no rule from a partition not yet considered could be added. Hence, the result of Theorem 3 directly follows from the results proved in [7] in the same way as shown in [10] for the min_1 transition mode. From the proof of Theorem 2 and the results proved in [7], also the following general computational completeness results for tissue P systems with hybrid transition modes for rule partitions follow:

Theorem 4. *For $X \in \{Ps, N\}$,*

$$\begin{aligned} XRE = {} & XO_*h_*tP_1\,(\{=1\})\,[noncoop] \\ & XO_*h_*tP_1\,(\{max, =1\})\,[noncoop]\,. \end{aligned}$$

7 Summary

In this paper we have introduced tissue P systems with hybrid transition modes for rule partitions. With noncooperative rules as well as with the maximally parallel transition mode for all partitions, we obtain a characterization of the extended tabled Lindenmayer systems, whereas with the $=1$-mode for 3 partitions or with the $=1$-mode for 2 partitions and the maximally parallel transition mode for one partition we already are able to generate any recursively enumerable set of natural numbers. As for (purely) catalytic P systems, the descriptional complexity, especially with respect to the number of partitions, of tissue P systems with hybrid transition modes for rule partitions able to generate any recursively enumerable set of (vectors of) natural numbers remains as a challenge for future research.

References

1. F. Bernardini, M. Gheorghe, N. Margenstern, S. Verlan, Networks of cells and Petri nets, in: M. A. Gutiérrez-Naranjo et al. eds., *Proc. Fifth Brainstorming Week on Membrane Computing*, Sevilla, 2007, 33–62.
2. M. Beyreder, R. Freund, Membrane systems using noncooperative rules with unconditional halting, in: D. W. Corne et al. eds., *Membrane Computing - 9th Intern. Workshop*, Revised Selected and Invited Papers, LNCS 5391, Springer, 2009, 129–136
3. G. Ciobanu, L. Pan, Gh. Păun, M.J. Pérez-Jiménez, P systems with minimal parallelism, *Theoretical Computer Science* **378** (1) (2007), 117–130.
4. E. Csuhaj-Varjú, Networks of language processors, *Current Trends in Theoretical Computer Science* (2001), 771–790.
5. E. Csuhaj-Varjú, J. Dessow, J. Kelemen, Gh. Păun, *Grammar Systems: A Grammatical Approach to Distribution and Cooperation* , Gordon and Breach Science Publishers, Amsterdam 1994.
6. J. Dassow, Gh. Păun, On the power of membrane computing, *Journal of Universal Computer Science* **5** (2) (1999), 33–49.
7. 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.
8. R. Freund, Gh. Păun, M.J. Pérez-Jiménez, Tissue-like P systems with channel states, *Theoretical Computer Science* **330** (2005), 101–116.
9. R. Freund, S. Verlan, A formal framework for P systems, in: G. Eleftherakis, P. Kefalas, Gh. Paun (Eds.), *Pre-proceedings of Membrane Computing, Intern. Workshop – WMC8*, Thessaloniki, Greece, 2007, 317–330.
10. R. Freund, S. Verlan, (Tissue) P systems working in the k-restricted minimally parallel derivation mode, in: E. Csuhaj-Varjú et al., eds, *Proc. Intern. Workshop on Computing with Biomolecules*, Österreichische Computer Gesellschaft, 2008, 43–52.
11. M.L. Minsky, *Computation – Finite and Infinite Machines*, Prentice Hall, Englewood Cliffs, NJ, 1967.
12. Gh. Păun, Computing with membranes, *J. of Computer and System Sciences* **61**, 1 (2000), 108–143.
13. Gh. Păun, *Membrane Computing. An Introduction*, Springer-Verlag, Berlin 2002.
14. G. Rozenberg, A. Salomaa (Eds.), *Handbook of Formal Languages* (3 volumes), Springer-Verlag, Berlin, 1997.
15. Gh. Păun, Y. Sakakibara, T. Yokomori, P systems on graphs of restricted forms, *Publicationes Matimaticae* **60**, 2002.
16. Gh. Păun, 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, American Mathematical Society, 1999, 217–232.
17. The P Systems web page: `http://ppage.psystems.eu`.