# (Tissue) P Systems Using Non-cooperative Rules Without Halting Conditions

Markus Beyreder, Rudolf Freund

Faculty of Informatics
Vienna University of Technology
Favoritenstr. 9, A-1040 Wien, Austria
E-mails: {markus,rudi}@emcc.at

**Summary.** We consider (tissue) P systems using non-cooperative rules, but considering computations without halting conditions. As results of a computation we take the contents of a specified output membrane/cell in each derivation step, no matter whether this computation will ever halt or not, eventually taking only results completely consisting of terminal objects only. The computational power of (tissue) P systems using non-cooperative rules turns out to be equivalent to that of (E)0L systems.

## 1 Introduction

In contrast to the original model of P systems introduced in [5], in this paper we only consider non-cooperative rules. Moreover, as results of a computation we take the contents of a specified output membrane in each derivation step, no matter whether this computation will ever halt or not, eventually taking only results completely consisting of terminal objects. In every derivation step, we apply the traditional maximal parallelism. Other derivation modes could be considered, too, but, for example, applying the sequential derivation mode would not allow us to go beyond context-free languages. As the model defined in this paper we shall take the more general one of tissue P systems (where the communication structure of the system is an arbitrary graph, e.g., see [4], [2]), which as a specific subvariant includes the original model of membrane systems if the communication structure allows for arranging the cells in a hierarchical tree structure.

The motivation to consider this specific variant of tissue P systems came during the Sixth Brainstorming Week in Sevilla 2008 when discussing the ideas presented in [3] with the authors Miguel Gutiérrez-Naranjo and Mario Pérez-Jiménez. They consider the evolution of deterministic (tissue) P systems with simple (i.e., non-cooperative) rules and aim to find a mathematically sound representation of such systems in order to deduce their behaviour and, on the other hand, to find suitable corresponding P systems for a given mathematical system with specific behaviour. Whereas in that paper only deterministic P systems are considered, which allows

for a mathematical representation like for deterministic 0L systems, and as well real values for the coefficients assigned to the symbols are allowed, in this paper we restrict ourselves to the non-negative integer coefficients commonly used in traditional variants of (tissue) P systems.

We shall prove that the computational power of extended tissue P systems using non-cooperative rules is equivalent to that of E0L systems when taking all results appearing in the specified output cell consisting of terminal objects only.

The present paper is organized as follows. Section 2 briefly recalls the notations commonly used in membrane computing and the few notions of formal language theory that will be used in the rest of the paper; in particular, we report the definition of (extended) Lindenmayer systems. Section 3 is dedicated to the definition of tissue P systems with non-cooperative rules working in the maximally parallel derivation mode. The computational power of these classes of (extended) tissue P systems is then investigated in Section 4 in comparison with the power of the corresponding classes of (extended) Lindenmayer systems. Some further remarks and directions for future research are discussed in the last section.

## 2 Preliminaries

We here recall some basic notions concerning the notations commonly used in membrane computing (we refer to [6] for further details and to [9] for the actual state of the art in the area of P systems) and the few notions of formal language theory we need in the rest of the paper (see, for example, [8] and [1], as well as [7] for the mathematical theory of L systems).

An alphabet is a finite non-empty set of abstract symbols. Given an alphabet $V$, by $V^*$ we denote the set of all possible strings over $V$, including the empty string $\lambda$. The length of a string $x \in V^*$ is denoted by $|x|$ and, for each $a \in V$, $|x|_a$ denotes the number of occurrences of the symbol $a$ in $x$. A multiset over $V$ is a mapping $M : V \longrightarrow \mathbb{N}$ such that $M(a)$ defines the multiplicity of $a$ in the multiset $M$ ($\mathbb{N}$ denotes the set of non-negative integers). Such a multiset can be represented by a string $a_1^{M(a_1)} a_2^{M(a_2)} \dots a_n^{M(a_n)} \in V^*$ and by all its permutations, with $a_j \in V$, $M(a_j) \geq 0$, $1 \leq j \leq n$. In other words, we can say that each string $x \in V^*$ identifies a finite multiset over $V$ defined by $M_x = \{(a, |x|_a) \mid a \in V\}$. Ordering the symbols in $V$ in a specific way, i.e., $(a_1, \dots, a_n)$ such that $\{a_1, \dots, a_n\} = V$, we get a Parikh vector $\left(|x|_{a_1}, \dots, |x|_{a_n}\right)$ associated with $x$. The set of all multisets over $V$ is denoted by $M_V$, the set of all Parikh vectors by $Ps\,(V^*)$. In the following, we shall not distinguish between multisets and the corresponding Parikh vectors. Given two multisets $x$ and $y$, with $x, y \in V^*$, we say that the multiset $x$ includes the multiset $y$, or the multiset $y$ is included in the multiset $x$, and we write $x \sqsupseteq y$, or $y \sqsubseteq x$, if and only if $|x|_a \geq |y|_a$, for every $a \in V$. The union of two multisets $x$ and $y$ is denoted by $x \sqcup y$ and is defined to be the multiset with $|x \sqcup y|_a = |x|_a + |y|_a$, for every $a \in V$. For $m, n \in \mathbb{N}$, by $[m..n]$ we denote the set $\{x \in \mathbb{N} \mid m \leq x \leq n\}$.

An extended Lindenmayer system (an E0L system for short) is a construct $G = (V, T, P, w)$, where $V$ is an alphabet, $T \subseteq V$ is the *terminal* alphabet, $w \in V^*$ is the *axiom*, and $P$ is a finite set of *non-cooperative rules* over $V$ of the form $a \to x$. In a derivation step, each symbol present in the current sentential form is rewritten using one rule arbitrarily chosen from $P$. The language generated by $G$, denoted by $L(G)$, consists of all the strings over $T$ which can be generated in this way by starting from $w$. An E0L system with $T = V$ is called a 0L system. As a technical detail we have to mention that in the theory of Lindenmayer systems usually it is required that for every symbol $a$ from $V$ at least one rule $a \to w$ in $P$ exists. If for every symbol $a$ from $V$ exactly one rule $a \to w$ in $P$ exists, then this Lindenmayer system is called *deterministic*, and we use the notations DE0L and D0L systems. By $E0L$ and $0L$ ($DE0L$ and $D0L$) we denote the families of languages generated by (deterministic) E0L systems and 0L systems, respectively. It is known from [8] that $CF \subset E0L \subset CS$, with $CF$ being the family of context-free languages and $CS$ being the family of context-sensitive languages, and that $CF$ and $0L$ are incomparable, with $\left\{a^{2^n} \mid n \geq 0\right\} \in D0L - CF$.

As the paper deals with P systems where we consider symbol objects, we will also consider E0L systems as devices that generate sets of (vectors of) non-negative integers; to this aim, given an E0L system $G$, we define the set of non-negative integers generated by $G$ as the length set $N(G) = \{\, |x| \mid x \in L(G) \,\}$ as well as $Ps\,(G)$ to be the set of Parikh vectors corresponding to the strings in $L\,(G)$. In the same way, the length sets and the Parikh sets of the languages generated by context-free and context-sensitive grammars can be defined. The corresponding families of sets of (vectors of) non-negative integers then are denoted by $NX$ and $PsX$, for $X \in \{E0L, 0L, DE0L, D0L, CF, CS\}$, respectively.

## 3 Tissue P Systems With Non-cooperative Rules

Now we formally introduce the notion of tissue P systems with non-cooperative rules by giving the following definition.

**Definition 1.** *An* extended tissue P system with non-cooperative rules *is a construct*

$$\Pi = (n, V, T, R, C_0, i_0)$$

*where*

1. *$n$ is the number of* cells*;*
2. *$V$ is a finite alphabet of symbols called* objects*;*
3. *$T \subseteq V$ is a finite alphabet of* terminal symbols *(*terminal objects*);*
4. *$R$ is a finite set of multiset rewriting rules of the form*

$$(a, i) \to (b_1, h_1) \ldots (b_k, h_k)$$

*for $i \in [1..k]$, $a \in V$ as well as $b_j \in V$ and $h_j \in [1..n]$, $j \in [1..k]$;*

5. $C_0 = (w_1, \ldots, w_n)$, *where the* $w_i \in V^*$, $i \in [1..n]$, *are finite multisets of objects for each* $i \in [1..n]$,
6. $i_0$ *is the* output cell.

A rule $(a, i) \to (b_1, h_1) \ldots (b_k, h_k)$ in $R_i$ indicates that a copy of the symbol $a$ in cell $i$ is erased and instead, for all $j \in [1..k]$, a copy of the symbol $b_j$ is added in cell $h_j$.

In any configuration of the tissue P system, a copy of the symbol $a$ in cell $i$ is represented by $(a, i)$, i.e., $(a, i)$ is an element of $V \times [1..n]$.

$\Pi$ is called *deterministic* if in every cell for every symbol from $V$ exactly one rule exists.

From the initial configuration specified by $(w_1, ..., w_n)$, the system evolves by transitions getting from one configuration to the next one by applying a maximal set of rules in every cell, i.e., by working in the *maximally parallel derivation mode*. A *computation* is a sequence of transitions. In contrast to the common use of P systems to generate sets of multisets, as a result of the P system we take the contents of cell $i_0$, provided it only consists of terminal objects only, at each step of any computation, no matter whether this computation will ever stop or not, i.e., we do not take into account any halting condition. The set of all multisets generated in that way by $\Pi$ is denoted by $L(\Pi)$. If we are only interested in the number of symbols instead of the Parikh vectors, the corresponding set of numbers generated by $\Pi$ is denoted by $N(\Pi)$.

The family of sets of multisets generated by tissue P systems with non-cooperative rules with at most $n$ cells in the maximally parallel derivation mode is denoted by $PsEtOP_n(noncoop, maxpar)$. Considering only the length sets instead of the Parikh vectors of the results obtained in the output cell during the computations of the tissue P systems, we obtain the family of sets of non-negative integers generated by tissue P systems with non-cooperative rules with at most $n$ cells in the maximally parallel derivation mode, denoted by $NEtOP_n(noncoop, maxpar)$. The corresponding families generated by non-extended tissue P systems – where all symbols are terminal – are denoted by $XtOP_n(noncoop, maxpar)$, $X \in \{Ps, N\}$. For all families generated by (extended) tissue P systems as defined before, we add the symbol $D$ in front of $t$ if the underlying systems are deterministic. If the number of cells is allowed to be arbitrarily chosen, we replace $n$ by $*$.

### 3.1 A well-known example

Consider the D0L system with the only rule $a \to aa$, i.e.,

$$G = (\{a\}, \{a\}, \{a \to aa\}, a).$$

As is well known, the language generated by $G$ is $\{a^{2^n} \mid n \geq 0\}$ and therefore $N(G) = \{2^n \mid n \geq 0\}$.

The corresponding deterministic one-cell tissue P system is

$$\Pi = (\{a\}, \{a\}, \{(a, 1) \to (a, 1)\,(a, 1)\}, (a)).$$

Obviously, we get $L(\Pi) = Ps(L(G))$ and $N(G) = N(\Pi)$.

We should like to point out that in contrast to this tissue P system without imposing halting, there exists no tissue P system with only one symbol in one cell

$$\Pi = (\{a\}, \{a\}, R, (w))$$

that with imposing halting is able to generate $\{2^n \mid n \geq 0\}$, because such systems can generate only finite sets (singletons or the empty set):

- if $w = \lambda$, then $N(\Pi) = \{0\}$;
- if $R$ is empty, then $N(\Pi) = \{|w|\}$;
- if $w \neq \lambda$ and $R$ contains the rule $a \to \lambda$, then $N(\Pi) = \{0\}$, because no computation can stop as long as the contents of the cell is not empty;
- if $w \neq \lambda$ and $R$ is not empty, but does not contain the rule $a \to \lambda$, then $R$ must contain a rule of the form $a \to a^n$ for some $n \geq 1$, yet this means that there exists no halting computation, i.e., $N(\Pi)$ is empty.

## 4 The Computational Power of Tissue P Systems With Non-cooperative Rules

In this section we present some results concerning the generative power of (extended) tissue P systems with non-cooperative rules; as we shall show, there is a strong correspondence between these P systems with non-cooperative rules and E0L systems.

**Theorem 1.** *For all $n \geq 1$,*

$$\begin{aligned} PsE0L &= PsEtOP_n(noncoop, maxpar) \\ &= PsEtOP_*(noncoop, maxpar). \end{aligned}$$

*Proof.* We first show that

$$PsE0L \subseteq PsEtOP_1(noncoop, maxpar):$$

Let $G = (V, T, P, w)$ be an E0L system. Then we construct the corresponding extended one-cell tissue P system

$$\Pi = (1, V, T, R, (w), 1)$$

with

$$R = \{(a, 1) \to (b_1, 1) \dots (b_k, 1) \mid a \to b_1 \dots b_k \in P\}.$$

Due to the maximal parallel derivation mode applied in the extended tissue P system $\Pi$, the derivations in $\Pi$ directly correspond to the derivations in $G$. Hence, $L(\Pi) = Ps(L(G))$.

As for all $n \geq 1$, by definition we have

$$PsEtOP_1 (noncoop, maxpar) \subseteq PsEtOP_n (noncoop, maxpar),$$

it only remains to show that

$$PsEtOP_* (noncoop, maxpar) \subseteq PsE0L :$$

Let

$$\Pi = (n, V, T, R, (w_1, \ldots, w_n), i_0)$$

be an extended tissue P system. Then we first construct the E0L system

$$G = (V \times [1..n], T_0, P, w)$$

with

$$w = \sqcup_{i=1}^{n} h_i (w_i)$$

($\sqcup$ represents the union of multisets) and

$$T_0 = h_{i_0} (T) \cup \cup_{j \in [1..n], j \neq i_0} h_j (V)$$

where the $h_i : V^* \to \{(a, i) \mid a \in V\}^*$ are morphisms with $h_i (a) = (a, i)$ for $a \in V$ and $i \in [1..n]$, as well as

$$P = R \cup P'$$

where $P'$ contains the rule $(a, i) \to (a, i)$ for $a \in V$ and $i \in [1..n]$ if and only if $R$ contains no rule for $(a, i)$ (which guarantees that in $P$ there exists at least one rule for every $b \in V \times [1..n]$).

We now take the projection $h : T_0^* \to T^*$ with $h ((a, i_0)) = a$ for all $a \in T$ and $h ((a, j)) = \lambda$ for all $a \in V$ and $j \in [1..n]$, $j \neq i_0$. Due to the direct correspondence of derivations in $\Pi$ and $G$, respectively, we immediately obtain $Ps (h (L (G))) = L (\Pi)$.

As $E0L$ is closed under morphisms (e.g., see [8], vol. 1, p. 266f.) and therefore $L (\Pi) = Ps (L (G'))$ for some E0L system $G'$, we finally obtain $L (\Pi) \in PsE0L$. □

As an immediate consequence of Theorem 1, we obtain the following results:

**Corollary 1.** *For all $n \geq 1$,*

$$\begin{aligned} NE0L &= NEtOP_n (noncoop, maxpar) \\ &= NEtOP_* (noncoop, maxpar). \end{aligned}$$

*Proof.* Given an E0L system $G$, we construct the corresponding extended tissue P system $\Pi$ as above in Theorem 1; then we immediately infer $N (G) = N (\Pi)$. On the other hand, given an extended tissue P system $\Pi$, by the constructions elaborated in Theorem 1, we obtain

$$N (\Pi) = N (G') = \{|x| \mid x \in h (L (G))\}$$

and therefore $N (\Pi) \in NE0L$. □

**Corollary 2.** *For* $X \in \{Ps, N\}$, $X0L = XtOP_1 (noncoop, maxpar)$.

*Proof.* This result immediately follows from the constructions elaborated in Theorem 1 with the specific restriction that for proving the inclusion $PstOP_1 (noncoop, maxpar) \subseteq Ps0L$ we can directly work with the symbols of $V$ from the given non-extended tissue P system $\Pi$ for the 0L system $G$ to be constructed (instead of the symbols from $V \times \{1\}$) and thus do not need the projection $h$ to get the desired result $L(\Pi) = L(G) \in Ps0L$. Besides this important technical detail, the results of this corollary directly follow from Theorem 1 and Corollary 1, because any non-extended system corresponds to an extended system where all symbols are terminal. □

For tissue P systems with only one cell, the non-cooperative rules can also be interpreted as antiport rules in the following sense: an antiport rule of the form $a/x$ in a single-cell tissue P system means that the symbol $a$ goes out to the environment and from there (every symbol is assumed to be available in the environment in an unbounded number) the multiset $x$ enters the single cell. The families of Parikh sets and length sets generated by (extended, non-extended) one-cell tissue P systems using antiport rules of this specific form working in the maximally parallel derivation mode are denoted by $XEtOP_1 (anti_{1,*}, maxpar)$ and $XtOP_1 (anti_{1,*}, maxpar)$ for $X \in \{Ps, N\}$, respectively. We then get the following corollary:

**Corollary 3.** *For* $X \in \{Ps, N\}$,

$$XEtOP_1 (anti_{1,*}, maxpar) = XE0L$$

*and*

$$XtOP_1 (anti_{1,*}, maxpar) = X0L.$$

*Proof.* The results immediately follow from the previous results and the fact that the application of an antiport rule $a/b_1 \dots b_k$ has exactly the same effect on the contents of the single cell as the non-cooperative evolution rule $(a, 1) \to (b_1, 1) \dots (b_k, 1)$. □

For one-cell tissue P systems, we obtain a characterization of the families generated by the deterministic variants of these systems by the families generated by the corresponding variants of Lindenmayer systems:

**Corollary 4.** *For* $X \in \{Ps, N\}$ *and* $Y \in \{noncoop, anti_{1,*}\}$,

$$XED0L = XtEDOP_1 (Y, maxpar)$$

*and*

$$XD0L = XtDOP_1 (Y, maxpar).$$

*Proof.* As already mentioned in the proof of Corollary 2, the results immediately follow from the constructions elaborated in Theorem 1 with the specific restriction that for proving the inclusion $PsEDtOP_1\,(noncoop, maxpar) \subseteq PsED0L$ we can directly work with the symbols of $V$ from the given (extended) deterministic tissue P system $\Pi$ for the ED0L system $G$ to be constructed (instead of the symbols from $V \times \{1\}$) and thus do not need the projection $h$ to get the desired result $L\,(\Pi) = L\,(G) \in PsED0L$. The remaining statements follow from these constructions in a similar way as the results stated in Corollaries 1, 2, and 3. □

The constructions described in the proofs of Corollary 2 and 4 cannot be extended to (non-extended, deterministic) tissue P systems with an arbitrary number of cells, because in that case again the application of a projection $h$ would be needed.

## 5 Conclusions and Future Research

In this paper we have shown that the Parikh sets as well as the length sets generated by (extended) tissue P systems with non-cooperative rules (without halting) coincide with the Parikh sets as well as the length sets generated by (extended) Lindenmayer systems.

In the future, we may also consider other variants of extracting results from computations in (extended) tissue P systems with non-cooperative rules, for example, variants of halting computations or only infinite computations, as well as other derivation modes as the sequential or the minimally parallel derivation mode. For the extraction of results, instead of the intersection with a terminal alphabet we may also use other criteria like the occurrence/absence of a specific symbol.

As inspired by the ideas elaborated in [3], we may investigate in more detail the evolution/behaviour of deterministic tissue P systems with non-cooperative rules based on the mathematical theory of Lindenmayer systems: as there is a one-to-one correspondence between deterministic tissue P systems with non-cooperative rules in one cell and D0L systems, the well-known mathematical theory for D0L systems can directly be used to describe/ investigate the behaviour of the corresponding deterministic tissue P systems with non-cooperative rules.

**Acknowledgements.** The authors gratefully acknowledge the interesting discussions with Miguel Gutiérrez-Naranjo and Mario Pérez-Jiménez on the ideas presented in their paper [3].

## References

1. J. Dassow and Gh. Păun, *Regulated Rewriting in Formal Language Theory*. EATCS Monograph in Theoretical Computer Science, Springer, Berlin, 1989.

2. R. Freund, Gh. Păun, and M.J. Pérez-Jiménez, Tissue-like P systems with channel states. *Theoretical Computer Science* **330** (2005), 101–116.
3. M.A. Gutiérrez-Naranjo and M.J. Pérez-Jiménez, Efficient computation in real-valued P systems, see this volume.
4. C. Martín-Vide, Gh. Păun, J. Pazos, and A. Rodríguez-Patón, Tissue P Systems, *Theoretical Computer Science*, **296** (2003), 295–326.
5. Gh. Păun, Computing with Membranes, *Journal of Computer and System Sciences*, **61** (2000), 108–143.
6. Gh. Păun, *Membrane Computing. An Introduction.* Natural Computing Series, Springer, Berlin, 2002.
7. G. Rozenberg and A. Salomaa, *The Mathematical Theory of L Systems.* Academic Press, New York, 1980.
8. G. Rozenberg and A. Salomaa (Eds), *Handbook of Formal Languages.* 3 volumes, Springer, Berlin, 1997.
9. The P Systems Web Page: `http://ppage.psystems.eu`.