

Doktori (PhD) értekezés tézisei

# Unconventional computing models

Kántor Kristóf

Témavezető: Dr. Vaszil György



DEBRECENI EGYETEM

Informatikai Tudományok Doktori Iskola

Debrecen, 2019

# Contents

<b>1</b>	<b>Unconventional computing models</b>	<b>3</b>
1.1	Introduction	3
1.2	Primary goals	6
1.3	Results	14
<b>2</b>	<b>Nem-hagyományos számítási modellek</b>	<b>17</b>
2.1	Bevezetés	17
2.2	Célkitűzések	21
2.3	Eredmények	29
<b>3</b>	<b>Publications of Kristóf Kántor</b>	<b>32</b>
<b>4</b>	<b>References</b>	<b>34</b>

# 1 Unconventional computing models

## 1.1 Introduction

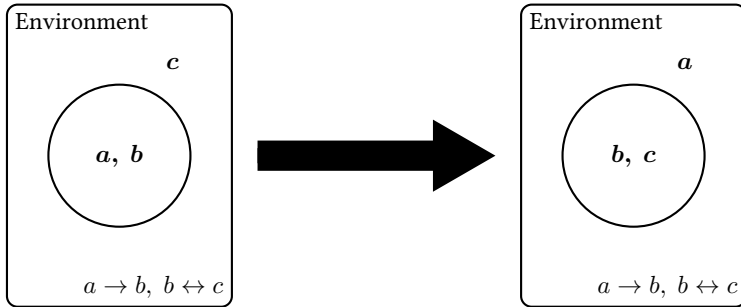
In the last few decades, unconventional computing became more and more popular as a research area. One of the main pioneers of unconventional computing, natural computing refers to computational processes observed in nature and to nature inspired, human-designed computing. Our understanding of the essence of computation is reinforced when analyzing different, complex natural phenomena in regard of computational processes. The principles and mechanisms underlying natural systems are the main characteristics for nature inspired, human-designed computing. Neural networks, quantum and molecular computing and many more are an integral part of natural computing.

Natural computing gave birth to membrane computing. Membrane systems were introduced in [13]. These so-called P systems are based on an abstraction, where the structure and operation of living cells are modeled. P systems are mainly observed in terms of computational power and complexity, comparing them to other P systems and systems from the field of classical computing.

Many efforts have been made to model information exchange and modification in its simplest forms between cells occurring in nature. During mathematical modeling, it is natural to use letters to symbolize objects that are used as information carrying agents in nature, for instance atoms, molecules or ions. P colonies represent a perfect example for the term “simplest form”, see [10, 11]. They are tissue-like membrane systems modeling cells and their shared environment. Considering information processing and storing capability, they are very limited: only a bounded number of objects may reside inside cells simultaneously. Agents are able to modify their descriptive information (objects) in two ways:

1. they are able to modify objects that are inside of them, resembling object evolution in nature (denoted by  $a \rightarrow b$ ),
2. they are able to exchange information with the environment, sending an object out and sucking another object in, resulting in an indirect communication between different agents (denoted by  $a \leftrightarrow b$ ).

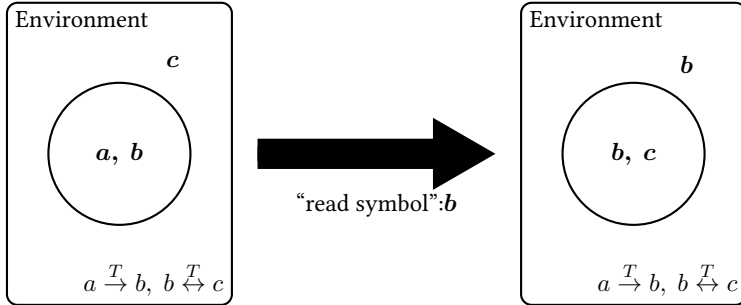
A visual example is given below:



When considering a discrete time-scale, applying rules in a maximal parallel way means that at every time-step every cell must apply a set of rules (called programs) that are applicable given the current contents of the device. Only those cells are allowed to remain “idle”, which have no applicable programs in the current configuration. As the result of a computation, it is natural to count the number of different objects that are present in the environment.

In [1], P colony automata have been born in order to describe sets of strings instead of numbers. By assuming the presence of an input tape with an input string in the environment, thus adding the tape version of existing types of rules in the form of  $b \xrightarrow{T} a$  or  $a \xleftarrow{T} b$ , it was possible to accept strings and string languages with the model. The new type of rules mean that they read a letter from

the input tape, but if more than one tape rule would be applied at once, they all must have read the same letter. To visualize the acceptance of strings, an example is given below:



Since only one letter can be read at every time-step, only those tape rules can be applied at the same time which read the same symbol from the input tape. This makes the model a bit complicated and in order to address this problem, a new model was introduced in [5] called generalized P colony automata or genPCol automata in short.

## 1.2 Primary goals

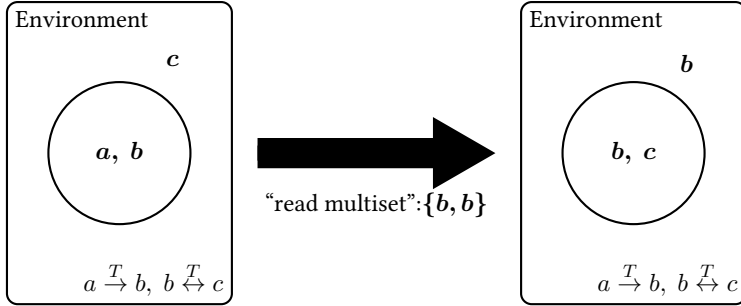
During this research, a new P system, called generalized P colony automata, or genPCol automata for short was introduced in [5] and later, extended in [6]. The main motivation was to create

a model that processes strings more naturally by associating them to the computations by keeping track of the communication with the environment instead of just reading input tapes, thus resembling other P systems more precisely, in particular P automata ([3, 4]). GenPCol automata combines the characteristics of both P automata and P colonies.

P automata are symport/antiport P systems ([12]) accepting string languages using a different approach than “ordinary” P colony automata. Instead of having input tapes and input strings, they keep track of their communications with the environment. Computing agents communicate freely with their environment without forcing themselves to a certain behaviour through a given input, where each object that can be requested for input by the communication rules are assumed to be available in an unbounded supply. In a single computational step, the automaton is allowed to read more than one symbol, that is, it reads multiset sequences over the course of computations. During computations, sequences of multisets can be determined that enter the system, thus characterizing the strings that are read by the P automaton.

This string processing behaviour is employed in genPCol au-

tomata, as illustrated in the following figure.



The computations describe accepted multiset sequences that are turned into accepted strings using a mapping function, that assigns strings to multisets over some alphabet  $\Sigma$ . Both non-tape and tape rules can be given to describe the operation of genPCol automata as in “ordinary” P colony automata, where tape rules imply the reading of the processed symbols over the course of a computation, however an automaton is allowed to read more than one symbol in a single computational step. The corollary to this is the complete nullification of conflicts between tape rules during computation since it is now possible to apply arbitrarily many tape rules in one time-step.

Formally, the model is presented below.

**Definition 1.** A *genPCol automaton* of capacity  $k$  and with  $n$  cells is a construct  $\Pi = (V, e, w_E, (w_1, P_1), \dots, (w_n, P_n), F)$ ,  $k, n \geq 1$ , where

- $V$  is an alphabet, the alphabet of the automaton, its elements are called objects;
- $e \in V$  is the environmental object of the automaton, that are always present in the environment in an infinite number;
- $w_E \in (V - \{e\})^*$  is a string representing the multiset of objects different from  $e$  which is found in the environment initially;
- $(w_i, P_i)$ ,  $1 \leq i \leq n$ , specifies the  $i$ -th cell where  $w_i$  is a multiset over  $V$ , it determines the initial contents of the cell, and its cardinality  $|w_i| = k$  is called the capacity of the system.  $P_i$  is a set of programs, each program is formed from  $k$  rules of the following types (where  $a, b, c \in V$ ):
  - tape rules of the form  $a \xrightarrow{T} b$ , or  $a \xleftrightarrow{T} b$ , called rewriting tape rules and communication tape rules, respectively;
  - or
  - nontape rules of the form  $a \rightarrow b$ , or  $c \leftrightarrow d$ , called

*rewriting (nontape) rules and communication (nontape) rules, respectively.*

*A program is called a tape program if it contains at least one tape rule.*

- *F is the set of accepting configurations of the automaton, that is a subset of configurations of the form  $(v_E, v_1, \dots, v_n)$ , where*
  - *$v_E \in (V - \{e\})^*$  denotes a multiset of objects different from  $e$  being in the environment, and*
  - *$v_i \in V^*$ ,  $1 \leq i \leq n$ , is the contents of the  $i$ -th cell.*

Now we give the definition for accepted input sequences and accepted string languages corresponding the model below.

**Definition 2.** *Let  $\Pi = (V, e, w_E, (w_1, P_1), \dots, (w_n, P_n), F)$  be a genPCol automaton. The set of input sequences accepted by  $\Pi$  is*

defined as

$$\begin{aligned}
A(\Pi) = & \{u_1 u_2 \dots u_s \mid u_i \in (V - \{e\})^*, 1 \leq i \leq s, \text{ and there is a} \\
& \text{configuration sequence } c_0, \dots, c_s, \text{ with} \\
& c_0 = (w_E, w_1, \dots, w_n), c_s \in F, \text{ and } c_i \Rightarrow c_{i+1} \text{ with} \\
& \bigcup_{p \in P_{c_i}} \text{read}(p) = u_{i+1} \text{ for all } 0 \leq i \leq s - 1\}.
\end{aligned}$$

As we have mentioned previously, in order for genPCol automata to accept string languages, a special mapping function is given.

**Definition 3.** Let  $\Pi$  be a genPCol automaton, let  $\Sigma$  be an alphabet and let  $f : (V - \{e\})^* \rightarrow 2^{\Sigma^*}$  be a mapping, that maps a multiset to a set of strings over  $\Sigma^*$ , moreover,  $f(u) = \varepsilon$  if and only if  $u$  is the empty multiset. The language accepted by  $\Pi$  with respect to  $f$  is defined as

$$L(\Pi, f) = \{f(u_1) \cdot f(u_2) \cdot \dots \cdot f(u_s) \in \Sigma^* \mid u_1 u_2 \dots u_s \in A(\Pi)\}.$$

Considering mapping functions, it is natural to think about every possible permutation of a given accepted multiset. Formally,  $f_{perm} : (V - \{e\})^* \rightarrow 2^{\Sigma^*}$ , where  $\Sigma = (V - \{e\})^*$ , and

$$f_{perm}(x) = \{y \in (V - \{e\})^* \mid y \in perm(x)\},$$

where  $perm(x) \subseteq V^*$  denotes the set of strings representing the multisets composed of the symbols of  $x$ , where each multiset is a permutation of  $x$ .

On the other hand, if a more generalized approach is required, functions from the class of finite transducers ( $\mathcal{F}_{TRANS}$ ) could be used to map accepted multisets to accepted strings. They assign a unique string to a given multiset, no matter its string representation. Formally, let  $V$  and  $\Sigma$  be two alphabets, and let  $\mathcal{M}_{FIN}(V) \subseteq \mathcal{M}(V)$  denote the set of finite subsets of the set of finite multisets over an alphabet  $V$ . Consider a mapping  $f : D \rightarrow 2^{\Sigma^*}$  for some  $D \in \mathcal{M}_{FIN}(V)$ . We say that  $f \in \mathcal{F}_{TRANS}$ , if for any  $v \in D$ , we have  $card(f(v)) = 1$ , and we can obtain  $f(v) = \{w\}$ ,  $w \in \Sigma^*$  by applying a deterministic finite transducer to any string representation of the multiset  $v$ , (as  $w$  is unique, the transducer must be constructed in such a way that all string representations of the multiset  $v$  as input result in the same  $w \in \Sigma^*$  as output, and moreover, as  $f$  should be nonerasing, the transducer produces a result with  $w \neq \varepsilon$  for any nonempty input).

There were three main aspects when defining further goals in regard of genPCol automata.

1. The first aspect relates to different questions regarding the computational power of the model:
  - a) First of all, what is the computational power of this model when restricting the capacity of the system, that is, the number of objects that may reside inside cells simultaneously?
  - b) How would the computational power change if we applied different semantic restrictions for the rules, such as com-tape, where the model always reads a multiset when communicating with the environment, or all-tape, where the model must read a multiset in every computational step?
  - c) What would the computational power be considering different classes of functions as mapping functions, such as the class of permutation mappings ( $\mathcal{F}_{perm}$ ), or the class of finite transducers ( $\mathcal{F}_{TRANS}$ )?
2. The second aspect relates to the comparison of genPCol automata to various non-classical computational models. How does the computational power compare to P automata given different restrictions (both semantic and capacity)?

3. The last aspect considers the practical utilization of the model, for instance, deterministic parsing. Can genPCol automata be defined in a way that it uses lookahead symbols? If so, how would it compare to context-free grammars using the  $LL(k)$  property?

### 1.3 Results

To give an answer to the primary goals of the research, the model was thoroughly analyzed in terms of computational power as well as the possibility of applications. Starting with the first aspect, the following results have been obtained (please note, that the theses are based on the references in the brackets, that refer the dissertation):

**Thesis 1.1** (Theorems 4.1, 4.4 and 4.5). *Considering genPCol automata with capacity one and no semantic restrictions to rules, they describe exactly the class of recursively enumerable languages.*

*This is true for both cases of classes of mapping functions, that is, for functions from  $\mathcal{F}_{perm}$  or  $\mathcal{F}_{TRANS}$ .*

*The same result holds for systems with at least capacity two having all-tape restriction and using mapping functions from  $\mathcal{F}_{TRANS}$ .*

**Thesis 1.2** (Theorem 4.2). *Using mapping functions from  $\mathcal{F}_{perm}$ , genPCol automata with capacity one and either all-tape or com-tape semantic rule restrictions are able to accept non-regular languages.*

**Thesis 1.3** (Theorems 4.3 and 4.8). *Considering mapping functions, no matter if functions from  $\mathcal{F}_{perm}$  or  $\mathcal{F}_{TRANS}$  are used, genPCol automata with com-tape semantic rule restriction can be simulated with one way, read-only input tape, restricted logarithmic space Turing machines.*

**Thesis 1.4** (Theorem 4.6). *The class of regular languages is a proper subset of languages recognized by genPCol automata with capacity one, using mapping functions from  $\mathcal{F}_{TRANS}$ .*

*This is true even if the genPCol automata have all-tape or com-tape semantic rule restrictions.*

To have an answer to the second aspect of questions, the following have been obtained.

**Thesis 2.1** (Corollary 4.2). *The class of languages accepted by genPCol automata using mapping functions from  $\mathcal{F}_{TRANS}$  and com-tape semantic rule restriction is a part of the class of languages accepted by P automata using the same mapping functions.*

**Thesis 2.2** (Theorem 4.7). *There are languages that cannot be recognized with  $P$  automata using mapping functions from the class  $\mathcal{F}_{perm}$ , but can be recognized with genPCol automata with capacity two, having com-tape semantic rule restriction and the same class of mapping functions.*

The last aspect of the primary goals has also been examined. The  $LL(k)$  property has been generalized for genPCol automata, and the following thesis holds true:

**Thesis 3** (Theorem 5.1). *Extended with the  $LL(1)$  property, genPCol automata are able to deterministically accept languages that cannot be generated by any context free  $LL(k)$  grammars ( $k \geq 1$ ).*

By defining theses, all the primary questions have been answered. The model was thoroughly analyzed in terms of computational power and practical utilization, focusing on different restrictions that were applied to the model.

## 2. Nem-hagyományos számítási modellek

### 2.1. Bevezetés

Az utóbbi néhány évtizedben a nem-hagyományos számítási modellek vizsgálata egyre elterjedtebb lett a kutatók körében. A fő úttörők közé tartoznak a természetes számítási modellek, melyek a természetben lejátszódó különböző megfigyelt folyamatokat modellezik, illetve a természet motiválta, ember által tervezett számítási modelleket jelentik.

A számítások lényegét jobban meg tudjuk érteni, ha számítási folyamatok különböző, komplex természeti jelenségeket elemezzük. A természetben előforduló rendszerek alapelvei és mechanizmusai a természet által inspirált, ember által tervezett számítási modellek fő jellemzői. A neurális hálózatok, a kvantum és a molekuláris számítási modellek és még sok más a természetes számítási modellek szerves része.

A természetes számítási modellek közé tartoznak a membrán rendszerek is, lásd [13]. A membrán rendszerek, más néven P rendszerek az élő sejtek struktúráján és működésén alapuló absztrakciók alapján kidolgozott számítási modellek. A P rendszereket főleg

számítási erő és számítási bonyolultság terén vizsgálják a kutatók, ahol egymással, illetve más, hagyományos számítási modellekkel is összehasonlításra kerülnek.

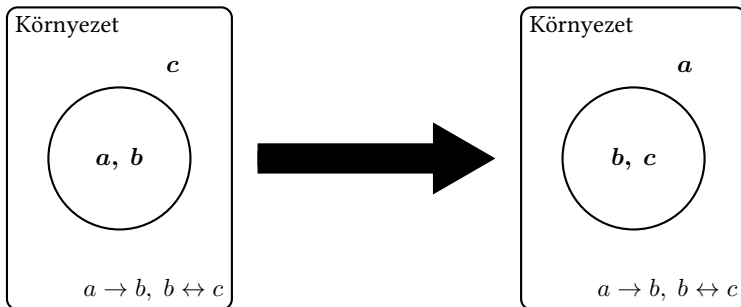
A kutatók sok erőfeszítést tettek, hogy minél egyszerűbb módon modellezzék a sejtek közötti információ áramlást és átalakulást. Matematikai modellezés során főleg betűket használunk olyan objektumok szimbolizálására, amelyek a természetben információ hordozó anyagként vannak jelen. Ilyen anyagok például az atomok, molekulák, ionok.

A P kolóniák [10, 11] tökéletes példaként szolgálnak arra, amikor egyszerűsége törekszünk a modellezés során. Ezek a szöveteszerű membrán rendszerek sejteket és a közöttük lévő közös környezetet modellezik. Ha az információ feldolgozási és tárolási képességeit vizsgáljuk a rendszernek, láthatjuk, hogy nagyon limitáltak: csak korlátozott számú objektum lehet a sejtekben egyidejűleg, továbbá az ágensek kétféle módon módosíthatják az objektumokat:

1. képesek a természetben megfigyelhető spontán, sejten belüli információ átalakulást elvégezni, melyet  $a \rightarrow b$  -vel jelölünk,

2. illetve képesek kommunikálni a környezettel, amit oly módon tesznek, hogy kiküldenek a környezetbe egy objektumot és egyidejűleg egy másikat beszipantanak, ezáltal megvalósítják az indirekt kommunikációt a többi ágens között (jelölése:  $a \leftrightarrow b$ ).

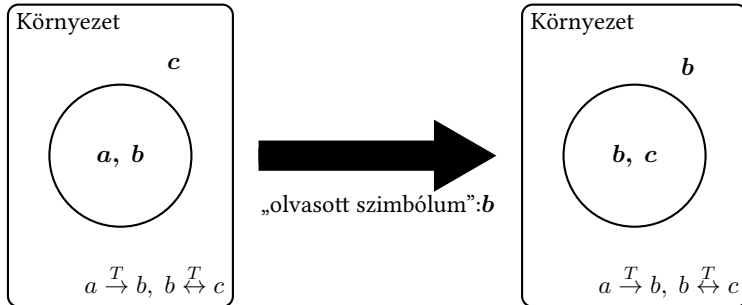
Egy példával szemléltetjük ezt a működést:



Diszkrét időskálát figyelembe véve a szabályok maximálisan párhuzamos alkalmazása azt jelenti, hogy minden ágensnek minden egyes lépésben szabályok egy halmazát (más nevén programot) kell alkalmaznia, ami alól csak azok lehetnek kivételek, melyek az adott konfigurációban egyik programjuk alkalmazására sem képesek. A számítás eredményeként a környezetben előforduló különböző objektumok számosságát értjük.

A P kolónia automaták [1] elmélete abból a célból született,

hogy karakterláncok halmazait legyünk képesek leírni számhalmazok helyett. Ebben a modellben egy bemeneti szalag és bemeneti karakterlánc létét feltételezzük a környezetben, továbbá a meglévő szabálytípusoknak szalagos változatát is bevezetjük a következő formákban:  $b \xrightarrow{T} a$  és  $a \xleftrightarrow{T} b$ . Ezzel a kiterjesztéssel lehetőség nyílt karakterláncok és karakterlánc nyelvek elfogadására. Az új típusú szabályok alkalmazása hatására az automata egy betűt olvas a bemeneti szalagról, viszont ha egyszerre több ilyen szabály kerülne alkalmazásra, akkor ezeknek a szabályoknak ugyanazt a betűt kell olvasniuk. A karakterek olvasása a következő ábrán jól megfigyelhető:



Mivel csak egy betűt képes olvasni egy lépésben az automata, csak azok a szalagszabályok alkalmazhatók egyszerre, amelyek ugyanazt a betűt olvassák a bemeneti szalagról. Ez bonyolulttá te-

szi a modellt, így a probléma kiküszöbölése céljából egy új modell került bevezetésre.

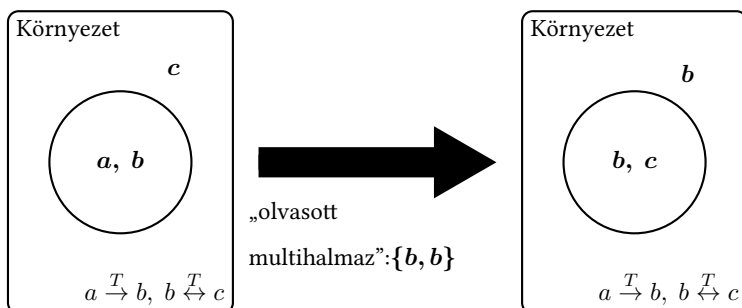
## 2.2. Célkitűzések

Kutatásaim folyamán egy újonnan bevezetett P rendszerrel, a generalizált P kolónia automatákkal, vagy más néven a genPCol automatákkal [5, 6] foglalkoztam. A dolgozat fő motivációja egy olyan P rendszer létrehozása volt, ahol a kerekterlanc feldolgozás input szalagok elolvasása helyett a környezettel való kommunikációs szabályok alkalmazásainak megjegyzésével valósul meg. Ez a viselkedés jobban hasonlít a többi P rendszerhez, különösen a P automatahoz ([3, 4]). A genPCol automata egyesíti mind a P automata és a P kolónia tulajdonságait.

A P automaták symport/antiport P rendszerek ([12]), amelyek különböző módon közelítik meg a karakterlanc nyelvek elfogadását, mint a „közönséges” P kolónia automaták. Bemeneti szalagok és karakterlancok helyett a környezettel való kommunikációra figyelnek, ahol bizonyos esetekben karakterlancokat olvashat az automata. A számítási ágensek szabadon kommunikálnak a környezettel, anélkül, hogy bármilyen viselkedésre lennének kényszerítve

egy input karakterlánc által. Egyetlen számítási lépésben egynél több szimbólumot is képesek olvasni, tehát egy számítási folyamat során multihalmazok sorozatát olvassák be, amikből leképzések segítségével meg lehet állapítani tényleges karakterlánc sorozatokat.

Ezt a sztring feldolgozó viselkedést alkalmazzák a generalizált P kolónia automaták, amint azt az alábbi ábra is szemlélteti.



Egy automatához definiált leképző függvény segítségével a számítások által leírt multihalmaz szekvenciák különböző karakterláncok sorozatává alakíthatóak. Ezek a leképző függvények karakterláncokat rendelnek multihalmazokhoz valamilyen  $\Sigma$  ábécé felett. Az automata viselkedését leíró szabályok lehetnek szalagosak és nem szalagosak. Mind a szalag és nem-szalag szabályokat megadhatjuk úgy a generalizált P kolónia automaták működésének meghatározásához, mint a „közönséges” P kolónia automaták

esetén, ahol a szalagos szabályok alkalmazásával lehetséges a számítási folyamat alatt megjegyezni és elfogadni különböző multihalmaz sorozatokat. A tetszőlegesen sok szalagszabály egy időben való alkalmazási lehetősége végett a szalagszabályok közötti konfliktusok teljes eltörlését értük el a modellel.

Formálisan az alábbi módon néz ki a modell.

**1. Definíció.** *Egy  $k \geq 1$  kapacitású és  $n \geq 1$  sejtből álló generalizált  $P$  kolónia automata alatt a következő konstrukciót értjük:*

$\Pi = (V, e, w_E, (w_1, P_1), \dots, (w_n, P_n), F)$ , ahol

- $V$  egy ábécé, az automata ábécéje, melynek elemeit objektumoknak nevezzük;
- $e \in V$  egy speciális, kijelölt környezeti objektuma az automatának, amiből minden pillanatban végtelen sok van a környezetben és mindig rendelkezésre áll;
- $w_E \in (V - \{e\})^*$  egy  $e$ -től különböző objektumokat tartalmazó, környezetben megtalálható multihalmaz, ami a kezdetben jelen van;
- $(w_i, P_i), 1 \leq i \leq n$ , leírja az  $i$ -edik sejtet, vagy más szóval ágenszt, ahol  $w_i$  egy  $V$  feletti multihalmaz, ami meghatároz-

za a sejt kezdeti tartalmát és  $|w_i| = k$  jelzi a rendszer kapacitását. Továbbá,  $P_i$  programok egy halmaza, ahol minden program  $k$  szabályt tartalmaz. A szabályok a következő alakúak lehetnek ( $a, b, c \in V$ ):

- szalagszabályok, azon belül is az újraíró szalag szabályok  $a \xrightarrow{T} b$  alakúak, a kommunikációs szalagszabályok pedig  $a \xleftrightarrow{T} b$  alakúak; vagy
- nem-szalag szabályok, azon belül is az újraíró (nem-szalag) szabályok  $a \rightarrow b$  alakúak, a kommunikációs (nem-szalag) szabályok pedig  $c \leftrightarrow d$  alakúak.

Egy programot szalagprogramnak nevezünk, ha legalább egy szalagszabályt tartalmaz.

- $F$  az automata elfogadó konfigurációinak a halmaza, ami a konfigurációk halmazának része. Egy konfiguráció alatt a  $(v_E, v_1, \dots, v_n)$   $n$ -est értjük, ahol
  - $v_E \in (V - \{e\})^*$  jelzi a környezet tartalmát az  $e$  ki-tüntetett objektum kivételével, és
  - $v_i \in V^*$ ,  $1 \leq i \leq n$  jelenti az  $i$ -edik sejt tartalmát.

Most megadjuk a modellhez tartozó elfogadott multihalmaz

szekvenciák és az elfogadott sztring nyelvek definícióit.

**2. Definíció.** Legyen  $\Pi = (V, e, w_E, (w_1, P_1), \dots, (w_n, P_n), F)$  egy generalizált  $P$  kolónia automata. A  $\Pi$  által elfogadott multihalmaz szekvenciák a következőképpen írhatók le:

$$A(\Pi) = \{u_1 u_2 \dots u_s \mid u_i \in (V - \{e\})^*, 1 \leq i \leq s, \text{ és van olyan } c_0, \dots, c_s \text{ konfigurációs sorozat, hogy}$$

$$c_0 = (w_E, w_1, \dots, w_n), c_s \in F, \text{ és } c_i \Rightarrow c_{i+1}, \text{ ahol}$$

$$\bigcup_{p \in P_{c_i}} \text{read}(p) = u_{i+1} \text{ minden } 0 \leq i \leq s - 1 \text{ esetén}\}.$$

Ahogy korábban említettük, egy speciális leképzésre van szükségünk ahhoz, hogy a generalizált  $P$  kolónia automata sztring nyelveket is el tudjon fogadni.

**3. Definíció.** Legyen  $\Pi$  egy generalizált  $P$  kolónia automata,  $\Sigma$  egy ábécé, valamint legyen  $f : (V - \{e\})^* \rightarrow 2^{\Sigma^*}$  egy olyan leképzés, ami egy multihalmazhoz  $\Sigma^*$  feletti sztringek halmazát rendeli, továbbá  $f(u) = \varepsilon$  akkor és csak akkor, ha  $u$  az üres multihalmaz.  $\Pi$  által elfogadott nyelv az  $f$  tekintetében a következő:

$$L(\Pi, f) = \{f(u_1) \cdot f(u_2) \cdot \dots \cdot f(u_s) \in \Sigma^* \mid u_1 u_2 \dots u_s \in A(\Pi)\}.$$

A lehetséges leképezések közül talán a legtermészetesebb egy multihalmaz összes lehetséges sztring permutációjára gondolni. A permutációs leképezéseket tartalmazó függvényosztály  $(\mathcal{F}_{perm})$  pontosan ezt gondolatot valósítja meg. Formálisan,  $f_{perm} : (V - \{e\})^* \rightarrow 2^{\Sigma^*} \in \mathcal{F}_{perm}$ , ahol  $\Sigma = (V - \{e\})^*$ , és  $f_{perm}(x) = \{y \in (V - \{e\})^* \mid y \in perm(x)\}$ , ahol  $perm(x) \subseteq V^*$  azon karakterláncok halmazát jelenti, amelynek elemei az  $x$  multihalmazból vett szimbólumokból képzett multihalmazok permutációi.

Másrésről, ha általánosabb leképezésekben gondolkodunk, a véges transzducereket tartalmazó függvényosztályból  $(\mathcal{F}_{TRANS})$  származó leképezéseket célszerű használni. Ezen transzducerek egyedi karakterláncokat rendelnek egy-egy adott multihalmazhoz, függetlenül a multihalmazok sztringreprezentációaiktól. Egy multihalmaznak a sztringreprezentációja alatt azt értjük, hogy hogyan tudjuk őket sztringgé alakítani úgy, hogy az elemek sorrendiségét is figyelembe vesszük.

Formálisan, legyen  $V$  és  $\Sigma$  két ábécé és jelölje  $\mathcal{M}_{FIN}(V) \subseteq \mathcal{M}(V)$  a  $V$  feletti véges multihalmazok véges részhalmazait. Vegyük figyelembe a következő leképezést:  $f : D \rightarrow 2^{\Sigma^*}$  valami-

lyen  $D \in \mathcal{M}_{FIN}(V)$ -re. Azt mondjuk,  $f \in \mathcal{F}_{TRANS}$ , ha bármely  $v \in D$ -re  $card(f(v)) = 1$  és  $f(v) = \{w\}$ ,  $w \in \Sigma^*$ , ha  $v$  bármely sztringreprezentációjára alkalmazunk egy véges determinisztikus transzducert. Mivel  $w$  egyedi, a transzducert úgy kell megalkotni, hogy a  $v$  multihalmaz összes sztringreprezentációjához ugyanaz a  $w \in \Sigma^*$  tartozzon, továbbá  $f$  nem lehet törölő, tehát  $w \neq \varepsilon$  bármely nemüres inputra.

A generalizált P kolónia automatákhoz tartozó további célok kitérését és kérdések feltevését három főbb szempont alapján csoportosítjuk.

1. Az első szempont alapján, amely a modell számítási erejével kapcsolatos kérdésekkel foglalkozik, több különböző aspektus fogalmazható meg:
  - a) Mi a számítási ereje a generalizált P kolónia automatáknak, ha megszorításokat teszünk a rendszer kapacitását illetően, vagyis ha megszorítjuk az objektumok számát, ami egy időben egyszerre a sejtekben lehet?
  - b) Hogy változna a számítási erő, ha különböző szemantikai megszorításokat szabnánk a szabályokra? Mi

történne, ha a modell minden olyan lépésben olvasna egy multihalmazt, amikor kommunikál a környezettel (com-tape)? Hogyan módosulna a számítási erő, ha a modell minden számítási lépésben olvasna valamit, attól függetlenül, hogy kommunikál-e a környezettel (all-tape)?

c) Hogyan alakulna a számítási ereje a generalizált P kolónia automatáknak, ha különböző függvényosztályokat vennénk alapul a leképezések használatakor, mint például a permutációs leképezéseket tartalmazó függvényosztályt ( $\mathcal{F}_{perm}$ ), vagy a véges transzducek osztályát ( $\mathcal{F}_{TRANS}$ )?

2. A második szempont a generalizált P kolónia automaták összehasonlítása más nem-klasszikus számítási modellekkel. Hogyan viszonyul a modell számítási ereje a P automaták számítási erejéhez, ha alapul vesszük a különböző szemantikai és kapacitás általi megszorításokat is?

3. Az utolsó szempont a modell gyakorlati felhasználásáról szól. Lehetne-e például determinisztikus elemzéshez hasz-

nálni a generalizált P kolónia automatákat és kiterjeszteni úgy a definíciótjukat, hogy a modell előre tudjon olvasni  $k$  szimbólumot? Ha igen, hogyan viszonyulna ez a kiterjesztett modell az  $LL(k)$  tulajdonsággal rendelkező környezet-független nyelvtanokhoz?

## 2.3. Eredmények

A modell számítási erőre és gyakorlati alkalmazásokra is kiterjedő alapos elemzésen esett át, hogy választ adhassunk a fentebb említett kérdésekre. Az első szempont alapján definiált kérdésekre a következő tézisek lettek kimondva (a tézis mellett zárójelben lévő referencia a disszertációra mutat):

**1.1. Tézis** (4.1, 4.4 és 4.5 Tételek). *Az egy kapacitású, szabályok szemantikus megszorítása nélküli generalizált P kolónia automaták pontosan a rekurzívan felsorolható nyelvek osztályát jellemzik.*

*Ez igaz akkor is, ha  $\mathcal{F}_{perm}$ , vagy  $\mathcal{F}_{TRANS}$  függvényosztályokból választunk leképezéseket.*

*Ez az eredmény igaz kettő kapacitású és all-tape szabálymegszorítással rendelkező rendszerek esetén is, amik viszont csak a  $\mathcal{F}_{TRANS}$  függvényosztályból használnak leképezéseket.*

**1.2. Tézis** (4.2 Tétel).  $\mathcal{F}_{perm}$  függvényosztályból választott leképezések esetén az egy kapacitású, all-tape vagy com-tape szabálymegszorítással rendelkező generalizált  $P$  kolónia automaták képesek nem-reguláris nyelveket is elfogadni.

**1.3. Tézis** (4.3 és 4.8 Tételek). Tetszőlegesen  $\mathcal{F}_{perm}$  vagy  $\mathcal{F}_{TRANS}$  függvényosztályokból választott leképezés esetén a com-tape szabálymegszorítással rendelkező generalizált  $P$  kolónia automaták szimulálhatók csak olvasható és egyirányú bemenő szalagos, megszorított logaritmikus táras Turing-gépekkel.

**1.4. Tézis** (4.6 Tétel). A reguláris nyelvek osztálya valódi részhalmaza az  $\mathcal{F}_{TRANS}$ -beli leképezéseket használó, megszorításmentes, all-tape vagy com-tape szabálymegszorítással rendelkező generalizált  $P$  kolónia automaták által felismerhető nyelvek osztályainak.

Hogy választ adjuk a második szempont alapján feltett kérdéseknek, a következő két tézis lett kimondva:

**2.1. Tézis** (4.2 Következmény).  $\mathcal{F}_{TRANS}$  függvényosztályból választott leképezések esetén a com-tape szabálymegszorítással rendelkező generalizált  $P$  kolónia automaták által felismerhető nyelvek osztálya része az ugyanilyen leképezéseket használó  $P$  automatákkal felismerhető nyelvek osztályának.

**2.2. Tézis** (4.7 Tétel). *Vannak olyan nyelvek, amelyeket  $\mathcal{F}_{perm}$ -ből származó leképezéseket használó  $P$  automatákkal nem lehet felismerni, de ugyanilyen leképezéseket használó, kettő kapacitású és com-tape szabálymegszorítással rendelkező generalizált  $P$  kolónia automatákkal igen.*

Az utolsó szempont alapján kitűzött célnak is eleget tettünk a környezetfüggetlen nyelvtanokra ismert  $LL(k)$  tulajdonság generalizált  $P$  kolónia automatákra való általánosításával és a következő tézis kimondásával.

**3. Tézis** (5.1 Tétel). *Az  $LL(1)$  tulajdonsággal kibővített generalizált  $P$  kolónia automaták képesek olyan nyelveket determinisztikusan elfogadni, amelyeket nem lehet egyetlen környezetfüggetlen,  $LL(k)$  tulajdonsággal rendelkező nyelvtannal sem generálni ( $k \geq 1$ ).*

A tézisek kimondásával választ adtunk az összes kérdésre, amelyek a célkitűzésben definiálva lettek. A generalizált  $P$  kolónia automata nem-hagyományos számítási modellre különböző, szintaktikai vagy szemantikai megszorítások által definiált esetek tüzetesen meg lettek vizsgálva számítási erő terén, továbbá a gyakorlati alkalmazhatóság is be lett mutatva.

### 3 Publications of Kristóf Kántor

#### International journal publications

1. K. Kántor, Gy. Vaszil: Generalized P colony automata. *Journal of Automata, Languages and Combinatorics* 19(1-4), 145–156 (2014)
2. K. Kántor, Gy. aszil: On the classes of languages characterized by generalized P colony automata. *Theoretical Computer Science* 724, 35–44 (2018)

#### International conference proceedings

3. E. Csuhaj-Varjú, K. Kántor, Gy. Vaszil: Deterministic Parsing with P Colony Automata. In: Graciani C., Riscos-Núñez A., Păun G., Rozenberg G., Salomaa A. (eds.), *Enjoying Natural Computing*. Volume 11270 of *Lecture Notes in Computer Science*, Springer, Cham, 88–98 (2018)
4. K. Kántor, Gy. Vaszil: On the classes of languages characterized by generalized P colony automata. In: *Proceedings of the Fourteenth Brainstorming Week on Membrane Com-*

*puting*, 231–246. ISBN: 978-84-946316-1-0 (2016)

5. K. Kántor, Gy. Vaszil: Variants of P Colony Automata. In: *Ninth Workshop on Non-Classical Models of Automata and Applications (NCMA 2017), Short Papers*, 17–24 (2017)
6. K. Kántor, Gy. Vaszil: Generalized P colony automata and their relation to P automata. In: *18th International Conference on Membrane Computing, CMC 2017, Bradford, UK, July 25-28, 2017, Revised Selected Papers*. Volume 10725 of *Lecture Notes in Computer Science*, Springer, 167–182 (2017)

## 4 References

- [1] L. Cienciala, L. Ciencialová, E. Csuhaj-Varjú, G. Vaszil: Pcol automata: Recognizing strings with P colonies. In: Martínez Del Amor, M.A., Păun, G., Pérez Hurtado, I., Riscos Nuñez, A. (eds.) *Eighth Brainstorming Week on Membrane Computing*, Sevilla, February 1-5, 2010, 65–76. Fénix Editora (2010)
- [2] E. Csuhaj-Varjú, K. Kántor, Gy. Vaszil: Deterministic Parsing with P Colony Automata. In: Graciani C., Riscos-Núñez A., Păun G., Rozenberg G., Salomaa A. (eds) *Enjoying Natural Computing. Lecture Notes in Computer Science, vol 11270*, 88–98. Springer, Cham (2018)
- [3] E. Csuhaj-Varjú, M. Oswald, Gy. Vaszil: P automata, in [14], chapter 6, 144–167.
- [4] E. Csuhaj-Varjú, Gy. Vaszil: P automata or purely communicating accepting P systems. In: Gh. Păun, G. Rozenberg, A. Salomaa, C. Zandron, eds., *Membrane Computing. International Workshop, WMC-CdeA 2002, Curtea de Arges, Romania, August 19-23, 2002, Revised Papers. LNCS 2597*, Springer Berlin Heidelberg, 2003, 219–233.

- [5] K. Kántor, Gy. Vaszil: Generalized P colony automata. *Journal of Automata, Languages and Combinatorics* 19(1-4), 145–156 (2014)
- [6] K. Kántor, Gy. Vaszil: On the classes of languages characterized by generalized P colony automata. *Theoretical Computer Science* 724, 35–44 (2018)
- [7] K. Kántor, Gy. Vaszil: On the classes of languages characterized by generalized P colony automata. In: *Proceedings of the Fourteenth Brainstorming Week on Membrane Computing*, 231–246. ISBN: 978-84-946316-1-0 (2016)
- [8] K. Kántor, Gy. Vaszil: Variants of P Colony Automata. In: *Ninth Workshop on Non-Classical Models of Automata and Applications (NCMA 2017), Short Papers*, 17–24 (2017)
- [9] K. Kántor, Gy. Vaszil: Generalized P colony automata and their relation to P automata. In: *18th International Conference on Membrane Computing, CMC 2017, Bradford, UK, July 25–28, 2017, Revised Selected Papers. Lecture Notes in Computer Science, vol. 10725*, 167–182. Springer (2017)
- [10] J. Kelemen, A. Kelemenová, Gh. Păun, Preview of P colonies: A biochemically inspired computing model, in: M. Bedau et

- al., eds., *Workshop and Tutorial Proceedings. Ninth International Conference on the Simulation and Synthesis of Living Systems (Alife IX)* (Boston Mass., 2004) 82–86.
- [11] A. Kelemenová: P colonies. In: Paun, G., Rozenberg, G., Salomaa, A. (eds.) *The Oxford Handbook of Membrane Computing*, 584–593. Oxford University Press, Inc. (2010)
- [12] A. Păun, G. Păun: The power of communication: P systems with symport/antiport. *New Generation Comput.* 20(3), 295–306 (2002)
- [13] G Păun: Computing with membranes. *Journal of Computer and System Sciences* 61(1), 2000, 108-143
- [14] G. Păun, G. Rozenberg, A. Salomaa, eds., *The Oxford Handbook of Membrane Computing* (Oxford University Press, 2010).



Registry number: DEENK/291/2019.PL  
Subject: PhD Publikációs Lista

Candidate: Kristóf Szilveszter Kántor  
Neptun ID: FC7PRP  
Doctoral School: Doctoral School of Informatics  
MTMT ID: 10044485

### List of publications related to the dissertation

#### Foreign language international book chapters (1)

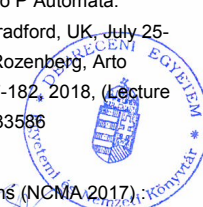
1. Csuhaj-Varjú, E., **Kántor, K. S.**, Vaszil, G.: Deterministic Parsing with P Colony Automata.  
In: Enjoying Natural Computing : Essays Dedicated to Mario de Jesús Pérez-Jiménez on the Occasion of His 70th Birthday. Eds.: Carmen Graciani, Agustín Riscos-Núñez, Gheorghe Păun, Grzegorz Rozenberg, Arto Salomaa, Springer, Cham, 88-98, 2018, (Lecture Notes in Computer Science, ISSN 0302-9743 ; 11270.) ISBN: 9783030002640

#### Foreign language scientific articles in international journals (2)

2. **Kántor, K. S.**, Vaszil, G.: On the classes of languages characterized by generalized P colony automata.  
*Theor. Comput. Sci.* 724, 35-44, 2018. ISSN: 0304-3975.  
DOI: <http://dx.doi.org/10.1016/j.tcs.2017.12.011>  
IF: 0.718
3. **Kántor, K. S.**, Vaszil, G.: Generalized P Colony Automata.  
*J. Autom. Lang. Comb.* 19 (1-4), 145-156, 2014. ISSN: 1430-189X.  
DOI: <http://dx.doi.org/10.25596/jalc-2014-145>

#### Foreign language conference proceedings (3)

4. **Kántor, K. S.**, Vaszil, G.: Generalized P Colony Automata and Their Relation to P Automata.  
In: Membrane Computing : 18th International Conference, CMC 2017, Bradford, UK, July 25-28, 2017 : Revised Selected Papers. Eds.: Marian Gheorghe, Grzegorz Rozenberg, Arto Salomaa, Claudio Zandron, Springer International Publishing, Cham, 167-182, 2018, (Lecture Notes in Computer Science, ISSN 0302-9743 ; 10725.) ISBN: 9783319733586
5. **Kántor, K. S.**, Vaszil, G.: Variants of P colony automata.  
In: Ninth Workshop on Non-Classical Models of Automata and Applications (NCMA 2017). Short Papers. Eds.: Rudolf Freund, Frantisek Mráz, Daniel Prusa, Institut für Computersprachen TU Wien, Wien, 17-24, 2017.





6. **Kántor, K. S.**, Vaszil, G.: On the classes of languages characterized by generalized P colony automata.

In: 14th Brainstorming Week on Membrane Computing. Eds.: Carmen Graciani, David Orellana-Martín, Agustín Riscos-Núñez, Álvaro Romero-Jiménez, Luis Valencia-Cabrera, Fénix Editora, Sevilla, 231-246, 2016. ISBN: 9788494631610

**Total IF of journals (all publications): 0,718**

**Total IF of journals (publications related to the dissertation): 0,718**

The Candidate's publication data submitted to the iDEa Tudóstér have been validated by DEENK on the basis of the Journal Citation Report (Impact Factor) database.

31 July, 2019





Nyilvántartási szám: DEENK/291/2019.PL  
Tárgy: PhD Publikációs Lista

Jelölt: Kántor Kristóf Szilveszter  
Neptun kód: FC7PRP  
Doktori Iskola: Informatikai Tudományok Doktori Iskola  
MTMT azonosító: 10044485

## A PhD értekezés alapjául szolgáló közlemények

### Idegen nyelvű, külföldi könyvrészletek (1)

1. Csuhaj-Varjú, E., **Kántor, K.**, Vaszil, G.: Deterministic Parsing with P Colony Automata.  
In: Enjoying Natural Computing : Essays Dedicated to Mario de Jesús Pérez-Jiménez on the Occasion of His 70th Birthday. Eds.: Carmen Graciani, Agustín Riscos-Núñez, Gheorghe Păun, Grzegorz Rozenberg, Arto Salomaa, Springer, Cham, 88-98, 2018, (Lecture Notes in Computer Science, ISSN 0302-9743 ; 11270.) ISBN: 9783030002640

### Idegen nyelvű tudományos közlemények külföldi folyóiratban (2)

2. **Kántor, K.**, Vaszil, G.: On the classes of languages characterized by generalized P colony automata.  
*Theor. Comput. Sci.* 724, 35-44, 2018. ISSN: 0304-3975.  
DOI: <http://dx.doi.org/10.1016/j.tcs.2017.12.011>  
IF: 0.718
3. **Kántor, K.**, Vaszil, G.: Generalized P Colony Automata.  
*J. Autom. Lang. Comb.* 19 (1-4), 145-156, 2014. ISSN: 1430-189X.  
DOI: <http://dx.doi.org/10.25596/jalc-2014-145>

### Idegen nyelvű konferencia közlemények (3)

4. **Kántor, K.**, Vaszil, G.: Generalized P Colony Automata and Their Relation to P Automata.  
In: Membrane Computing : 18th International Conference, CMC 2017, Bradford, UK, July 25-28, 2017 : Revised Selected Papers. Eds.: Marian Gheorghe, Grzegorz Rozenberg, Arto Salomaa, Claudio Zandron, Springer International Publishing, Cham, 167-182, 2018, (Lecture Notes in Computer Science, ISSN 0302-9743 ; 10725.) ISBN: 9783319733586
5. **Kántor, K.**, Vaszil, G.: Variants of P colony automata.  
In: Ninth Workshop on Non-Classical Models of Automata and Applications (NCMA 2017). Short Papers. Eds.: Rudolf Freund, Frantisek Mráz, Daniel Prusa, Institut für Computersprachen TU Wien, Wien, 17-24, 2017.





6. **Kántor, K.**, Vaszil, G.: On the classes of languages characterized by generalized P colony automata.

In: 14th Brainstorming Week on Membrane Computing. Eds.: Carmen Graciani, David Orellana-Martín, Agustín Riscos-Núñez, Álvaro Romero-Jiménez, Luis Valencia-Cabrera, Félix Editora, Sevilla, 231-246, 2016. ISBN: 9788494631610

**A közlő folyóiratok összesített impakt faktora: 0,718**

**A közlő folyóiratok összesített impakt faktora (az értekezés alapjául szolgáló közleményekre):  
0,718**

A DEENK a Jelölt által az IDEa Tudóstérbe feltöltött adatok bibliográfiai és tudományometriai ellenőrzését a tudományos adatbázisok és a Journal Citation Reports Impact Factor lista alapján elvégezte.

Debrecen, 2019.07.31.

