Generalized P Colony Automata

Multiset Processing with Systems of Simple Components with Complex Behavior

Kristof Kantor
University of Debrecen
• **Multisets**: collection of objects/symbols, multiplicities

• **Complex behavior**: computational completeness, universality

• **Simple building blocks**: simple symbol processing agents in a shared environment (multiset) which they modify
A “chemical style” approach to the notion of computation

<table>
<thead>
<tr>
<th>data</th>
<th>substances or molecules</th>
</tr>
</thead>
<tbody>
<tr>
<td>processing</td>
<td>chemical reaction</td>
</tr>
<tr>
<td>algorithm</td>
<td>substances and their reaction laws</td>
</tr>
</tbody>
</table>

- **Data structure:** multisets
- **Computation:** multiset transformation/processing
Outline

- P colonies
  - structure, functioning, computational power, multiset languages

- P colony automata
  - languages of strings of symbols

- Generalized P colony automata
  - languages of strings/sequences of multisets
P colonies

- A population of very **simple cells** in a **shared environment**:
  - **Fixed number** of objects (1, 2, 3) inside each cell
  - **Simple** rules (programs) for **moving** and **changing** the objects

- The objects are **exchanged** directly only between the **cells** and the **environment**

[Kelemen, Kelemenova, Paun 2004]
P colonies

\[ a \rightarrow b, c \leftrightarrow d \]
The computation

- Start in an **initial configuration**
- Apply the programs (sets of rules) in **parallel** in the cells, **halt** if no program is applicable
- The **result** is the **number** of the **multiplicities** of certain objects found in the **environment**
The computation

initial configuration

⇒ ... ⇒

a possible result
The computation

\[(e \rightarrow b, b \leftrightarrow e)\]
\[(e \rightarrow d, b \leftrightarrow e)\]

We obtain \(a^n c^n, n \geq 1\) in the environment.
P colony automata

- Response to the **changes in the environment**
- **Automata-like** behavior - an **input string** is given
- **Tape rules** and **non-tape rules**: the application of programs with tape rules reads a **symbol** of the input

[CIencialova, CIenciala, CsuhaJ-Varjú, Kelemenova, Vaszil 2010]
P colony automata

The effect of tape rules:

\[
\begin{align*}
g & \quad sdade \\
\Rightarrow \\
\begin{array}{c}
a \\
c
\end{array} & \quad \Rightarrow \\
\begin{array}{c}
s \\
c
\end{array} & \quad \quad \quad \quad \quad \\
\quad \quad \quad \quad \quad \\
\begin{array}{c}
g
\end{array} & \quad \quad \quad \quad \\
\begin{array}{c}
dade
\end{array}
\end{align*}
\]
Different computational modes...

...with different uses of the tape rules:

- **t-transition**, denoted by $\Rightarrow_t$, if $u' = u$ and $P_c$ is maximal set of programs with respect to the property that every $p \in P_c$ is a tape program with $\text{read}(p) = a$;

- **tmin-transition**, denoted by $\Rightarrow_{t\text{min}}$, if $u' = u$ and $P_c$ is maximal set of programs with at least one $p \in P_c$, such that $p$ is a tape program with $\text{read}(p) = a$;

- **tmax-transition**, denoted as $\Rightarrow_{t\text{max}}$, if $u' = u$ and $P_c = P_T \cup P_N$ where $P_T$ is a maximal set of applicable tape programs with $\text{read}(p) = a$ for all $p \in P_T$, the set $P_N$ is a set of nontape programs, and $P_c = P_T \cup P_N$ is maximal;

- **n-transition**, denoted by $\Rightarrow_n$, if $u' = au$ and $P_c$ is maximal set of nontape programs.
Power of the different modes

- **nt, ntmax, ntmin**: any recursively enumerable language can be accepted/characterized [Ciencialova, Cienciala, Csuhaj-Varjú, Kelemenova, Vaszil 2010]

- **t, one cell**: only CS languages can be generated [Cienciala, Ciencialova 2011a]

- **initial**: any recursively enumerable language can be characterized [Cienciala, Ciencialova 2011b]
Generalized P colony automata

- A maximal parallel set of programs is chosen
- The tape rules might “read” several different symbols (multiset) in one step.
- The set of input sequences accepted by a GenPCol: The set of the sequences of read multisets
The **language accepted** by a GenPCol in respect to a mapping \( f: (V - \{e\})^* \rightarrow 2^{\Sigma^*} \):

\[
\mathcal{L}(\Pi, f) = \{ f(u_1) \cdots f(u_s) \in \Sigma^* | u_1 \cdots u_s \text{ is an accepted input sequence} \}
\]
\( f_{\text{perm}} \) 

- \( f_{\text{perm}} : (V - \{e\})^* \rightarrow 2^{(V - \{e\})^*} \), where \( f(x) = \{ y \in (V - \{e\})^* | y = \text{perm}(x) \} \)
Generalized P colony automata modes using $f_{perm}$

- **All-tape**: all programs contain at least one tape rule
- **Com-tape**: all communication rules are tape rules
- **No restriction** (noted by *)

[Kántor, Vaszil 2013]
Turing machines with restricted space bound

A nondeterministic Turing machine with a one-way input tape is restricted $S(n)$ space bounded if the number of nonempty cells on the worktape(s) is bounded by $S(d)$, where $d$ is the distance of the reading head from the left-end of the one-way input tape.
A Turing machine with SPACEBOUND(n)

The length of the available worktape is bounded by the length of the input:
Turing machines with *restricted* space bound

1. After reading $d_1$ input cells:
Turing machines with restricted space bound

2. After reading $d_2$ input tape cells:
Computational power

- $\mathcal{L}(\text{GenPCol}, \text{com} - \text{tape}, f_{\text{perm}}) \cup \mathcal{L}(\text{GenPCol}, \text{all} - \text{tape}, f_{\text{perm}}) \subseteq \mathcal{L}(\text{GenPCol}, \ast, f_{\text{perm}})$

- $\mathcal{L}(\text{GenPCol}, \text{com} - \text{tape}, f_{\text{perm}}) \cap \mathcal{L}(\text{GenPCol}, \text{all} - \text{tape}, f_{\text{perm}}) - \mathcal{L}(\text{CF}) \neq \emptyset$

- $\mathcal{L}(\text{REG}) \subset \mathcal{L}(\text{GenPCol}, \text{all} - \text{tape}, f_{\text{perm}}) \cup \mathcal{L}(\text{GenPCol}, \text{com} - \text{tape}, f_{\text{perm}}) \cup \mathcal{L}(\text{GenPCol}, \ast, f_{\text{perm}})$
New results: prerequisite knowledge

- For every language $L \subseteq V^*$, $L \in LP$, which is not regular there is a string $w \in L$ which can be written in the form $w = w_1 ab w_2$, for some $w_1, w_2 \in V^*$ and $a, b \in V$ such that $w_1 b a w_2 \in L$.

[Freund, Kogler, Paun, Pérez-Jiménez, 2009]
New results

- $\{(ab)^n(cd)^n \mid n \geq 1\}$ can be accepted by a GenPCol in all-tape mode using $f_{\text{perm}}$.

- Proof: Let us consider the following GenPCol (1 cell, 2 capacity):

New results

\[
\begin{align*}
\{ x \rightarrow a, e \leftrightarrow e \}, \\
\{ e \rightarrow b, a \leftrightarrow e \}, \\
\{ b \rightarrow a, e \leftrightarrow e \}, \\
\{ b \rightarrow c, e \leftrightarrow e \}, \\
\{ c \rightarrow d, e \leftrightarrow a \}, \\
\{ d \leftrightarrow e, a \rightarrow c \}
\end{align*}
\]
New results, open problems summary

- Acceptance of \( \{(ab)^n(cd)^n \mid n \geq 1\} \) by a GenPCol in all-tape mode using \( f_{perm}(\Pi) \Rightarrow \Pi \) is able to accept a language that P automata with input mapping \( f_{perm} \) cannot

- Open question: \( \mathcal{L}(GenPCol, all-\text{tape}, f_{perm}) \supset \mathcal{L}(PA, f_{perm}) \) ?

- Open question: Power comparison of all-tape and com-tape modes?

- Open question: Computational power using other mapping functions?
Thank you for your attention!