Tissue P Systems with Protein on Cells

Bosheng Song, Mario J. Pérez-Jiménez, Linqiang Pan

Huazhong University of Science and Technology, Wuhan, China
University of Sevilla, Sevilla, Spain

Email: boshengsong@163.com

Outline

Biological background

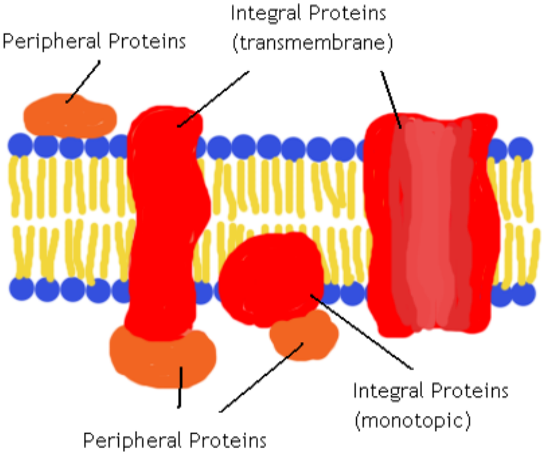
Tissue P systems with protein on cells

Universality

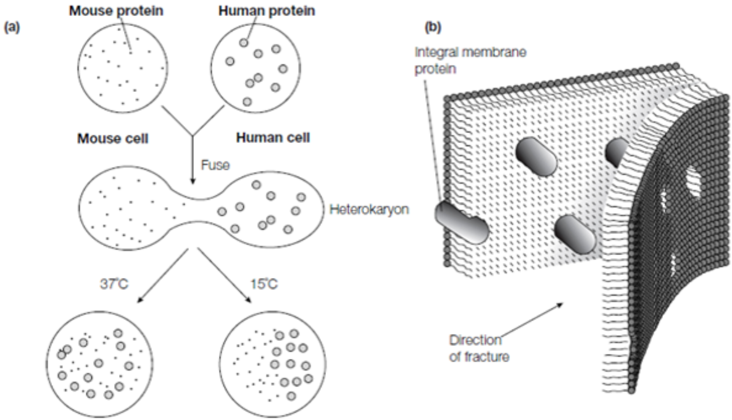
Computational efficiency

Open problems

Biological background



Biological background



Tissue P systems with protein on cells

Definition

A tissue P system with protein on cells of degree $q \geq 1$ is a tuple $\Pi = (\Gamma, P, \mathcal{E}, \mathcal{M}_1/p_1, \ldots, \mathcal{M}_q/p_q, \mathcal{R}, i_{out})$, where:

- Γ and P are finite non-empty alphabets such that $\Gamma \cap P = \emptyset$;
- $\mathcal{E}$ is a finite set of objects, such that $\mathcal{E} \subseteq \Gamma$;
- $\mathcal{M}_i$, $1 \leq i \leq q$, are finite multisets over Γ;
- p_i, $1 \leq i \leq q$, are elements in P (*there is one and only one copy of protein on each cell*);

Tissue P systems with protein on cells

- $i_{out} \in \{0, 1, \ldots, q\}$;
- $\mathcal{R}$ is a finite set of rules of the following forms:
  - Communication rules:
    (a) $(i, (p_i, u)/(p_j, v), j)$, for $i, j \in \{1, \ldots, q\}$, $i \neq j$, $p_i, p_j \in P$, $u, v \in \Gamma^*$.

Tissue P systems with protein on cells

- $i_{out} \in \{0, 1, \dots, q\}$;
- $\mathcal{R}$ is a finite set of rules of the following forms:
  - Communication rules:

    (a) $(i, (p_i, u)/(p_j, v), j)$, for $i, j \in \{1, \dots, q\}$, $i \neq j$, $p_i, p_j \in P$, $u, v \in \Gamma^*$.

    Both the protein p_i and the multiset u of objects are sent from region i to region j, and simultaneously, the protein p_j and the multiset v of objects are sent from region j to region i.

Tissue P systems with protein on cells

(b) $(i, (p_i, u)/v, 0)$, for $i \in \{1, \ldots, q\}$, $p_i \in P$, $u, v \in \Gamma^*$, $|uv| > 0$.

Tissue P systems with protein on cells

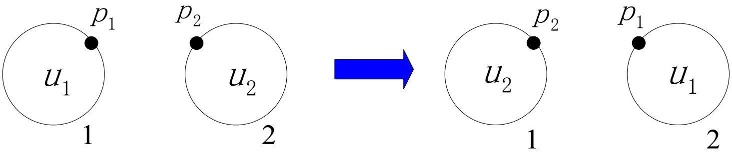
(b) $(i, (p_i, u)/v, 0)$, for $i \in \{1, \ldots, q\}$, $p_i \in P$, $u, v \in \Gamma^*$, $|uv| > 0$.
The multiset u of objects is sent from region i to the environment, and simultaneously, the multiset v of objects is sent from the environment to region i.

Tissue P systems with protein on cells

(b) $(i, (p_i, u)/v, 0)$, for $i \in \{1, \ldots, q\}$, $p_i \in P$, $u, v \in \Gamma^*$, $|uv| > 0$.
The multiset u of objects is sent from region i to the environment, and simultaneously, the multiset v of objects is sent from the environment to region i.

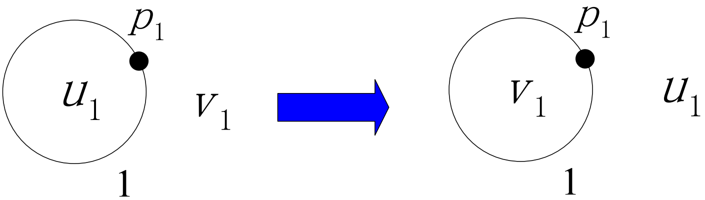
Note that when objects are communicated between a cell and the environment, the protein placed on that cell cannot be moved.

Example 1–Communication between two cells



$$(1, (p_1, u_1) / (p_2, u_2), 2)$$

Example 2–Communication between a cell and the environment



$$(1, (p_1, u_1) / v_1, 0)$$

Tissue P systems with protein on cells

The length of a communication rule is the total number of objects and proteins involved in that rule, that is, the length of rule $(i, (p_i, u)/(p_j, v), j)$ (resp., $(i, (p_i, u)/v, 0)$) is defined as $|u + v + 2|$ (resp., $|u + v + 1|$).

Semantics

- non-deterministic maximally parallel[1]:

  at each step, a set of applicable multiset of rules which is maximal in the sense that no further rule can be added being applicable.

[1]Gh. Păun, Computing with membranes, *Journal of Computer and System Sciences*, 61, 108–143, 2000

Some differences between cell-like P systems with proteins on membranes[1] and tissue-like P systems with protein on cells:

	cell-like	tissue-like
number of proteins	multiset	one and only one
evolved	both protein and objects	neither protein nor objects
place of proteins	never leave their membranes	move to other cells
number of objects	one inside and/or one outside	two multisets

[1] A. Păun, B. Popa, P systems with protein on membranes. *Fundamenta Informaticae* 72 (2006) 467–483.

Universality

Theorem

$NOP_2(commu_4) = NRE.$

Proof. The universality result is obtained by simulating register machines, which are a useful tool to characterize *NRE*[1].

We only have to prove the inclusion $NRE \subseteq NOP_2(commu_4)$.

Let $M = (m, H, l_0, l_h, I)$ be a register machine. We construct the P system Π to simulate register machine M.

[1]M.L. Minsky, *Computation: Finite and Infinite Machines*, Prentice–Hall, New Jersey, 1967.

$$\Pi = (\Gamma, P, \mathcal{E}, \mathcal{M}_1/p_1, \mathcal{M}_2/p_2, \mathcal{R}, i_{out}),$$

where:

- $\Gamma = \{a_r \mid 1 \leq r \leq m\} \cup \{l, l', l'', l''', l^{iv}, l^{v}, \bar{l} \mid l \in H\}$;
- $P = \{p_1, p_2\}$;
- $\mathcal{E} = \Gamma$;
- $\mathcal{M}_1 = \{l_0\}, \mathcal{M}_2 = \emptyset$;
- $i_{out} = 1$;

The set R of rules constructed as follows:

- For each ADD instruction $l_i : (\mathtt{ADD}(r), l_j, l_k)$, we introduce in R the rules

$$r_1 \equiv (1, (p_1, l_i)/l_j a_r, 0);$$
$$r_2 \equiv (1, (p_1, l_i)/l_k a_r, 0).$$

- For each SUB instruction $l_i : (\texttt{SUB}(r), l_j, l_k)$, we introduce in R the rules

$$r_3 \equiv (1, (p_1, l_i)/l_i' l_i'', 0);$$
$$r_4 \equiv (1, (p_1, l_i')/(p_2, \lambda), 2);$$
$$r_5 \equiv (1, (p_2, l_i'' a_r)/l_i''', 0);$$
$$r_6 \equiv (2, (p_1, l_i')/l_i^{iv}, 0);$$
$$r_7 \equiv (1, (p_2, l_i'')/(p_1, l_i^{iv}), 2);$$
$$r_8 \equiv (1, (p_2, l_i''')/(p_1, l_i^{iv}), 2);$$
$$r_9 \equiv (1, (p_1, l_i^{iv})/l_i^{v}, 0);$$

$$r_{10} \equiv (2, (p_2, l_i''')/\bar{l}_j, 0);$$

$$r_{11} \equiv (2, (p_2, l_i'')/\bar{l}_k, 0);$$

$$r_{12} \equiv (1, (p_1, l_i^v)/(p_2, \bar{l}_j), 2);$$

$$r_{13} \equiv (1, (p_1, l_i^v)/(p_2, \bar{l}_k), 2);$$

$$r_{14} \equiv (1, (p_2, \lambda)/(p_1, l_i^v), 2);$$

$$r_{15} \equiv (1, (p_1, l_i^v \bar{l}_j)/l_j, 0);$$

$$r_{16} \equiv (1, (p_1, l_i^v \bar{l}_k)/l_k, 0).$$

Table: For a SUB instruction $l_i : (\texttt{SUB}(r), l_j, l_k)$, where there is **at least one copy of object** a_r in cell 1. Let $z \in \{a_1, \ldots, a_m\}^*$, $z = a_r z'$

Step	Rules	Cell 1		Cell 2	
		Protein	Objects	Protein	Objects
0	–	p_1	$l_i z$	p_2	–
1	r_3	p_1	$l_i' l_i'' z$	p_2	–
2	r_4	p_2	$l_i'' z$	p_1	l_i'
3	r_5, r_6	p_2	$l_i''' z'$	p_1	l_i^{iv}
4	r_8	p_1	$l_i^{iv} z'$	p_2	l_i'''
5	r_9, r_{10}	p_1	$l_i^{v} z'$	p_2	$\bar{l}_j$
6	r_{12}	p_2	$\bar{l}_j z'$	p_1	l_i^{v}
7	r_{14}	p_1	$l_i^{v} \bar{l}_j z'$	p_2	–
8	r_{15}	p_1	$l_j z'$	p_2	–

Table: For a SUB instruction $l_i : (\mathtt{SUB}(r), l_j, l_k)$, where there is **no copy of object** a_r in cell 1. Let $z \in \{a_1, \ldots, a_m\}^*$, $a_r \notin z$

Step	Rules	Cell 1		Cell 2	
		Protein	Objects	Protein	Objects
0	–	p_1	$l_i z$	p_2	–
1	r_3	p_1	$l_i' l_i'' z$	p_2	–
2	r_4	p_2	$l_i'' z$	p_1	l_i'
3	r_6	p_2	$l_i'' z$	p_1	l_i^{iv}
4	r_7	p_1	$l_i^{iv} z$	p_2	l_i''
5	r_9, r_{11}	p_1	$l_i^v z$	p_2	$\bar{l}_k$
6	r_{13}	p_2	$\bar{l}_k z$	p_1	l_i^v
7	r_{14}	p_1	$l_i^v \bar{l}_k z$	p_2	–
8	r_{16}	p_1	$l_k z$	p_2	–

When the object l_h appears in cell 1, the computation stops.

The number of copies of a_1 in cell 1 clearly corresponds to the value of register 1 of M, hence $N(M) = N(\Pi)$.

Computational efficiency

Definition

A tissue P system with protein on cells and cell division of degree $q \geq 1$ is a tuple $\Pi = (\Gamma, P, \mathcal{E}, \mathcal{M}_1/p_1, \ldots, \mathcal{M}_q/p_q, \mathcal{R}, i_{out})$, and $\mathcal{R}$ also contains division rules of the form:

(c) $[\ p_i \mid a\ ]_i \rightarrow [\ p_i' \mid b\ ]_i [\ p_i'' \mid c\ ]_i$, for $i \in \{1, 2, \ldots, q\}$, $p_i, p_i', p_i'' \in P$, $a, b, c \in \Gamma$, $i \neq i_{out}$.

Solving the SAT problem

Theorem

The SAT *problem can be solved by using cell division and communication rules with length at most 4.*

Proof. The solution follows a brute force algorithm.

- *Generation phase*: all truth assignments for the n variables are produced (from r_1 to r_{10}).
- *Checking phase*: it is checked whether or not there is a truth assignment that makes the Boolean formula evaluate to be true (from r_{11} to r_{18}).
- *Output phase*: the system sends to the environment the right answer (from r_{19} to r_{24}).

For each $m, n \in \mathbb{N}$, we consider the recognizer tissue P system

$$\Pi(\langle m, n \rangle) = (\Gamma, P, \Sigma, \mathcal{E}, \mathcal{M}_1/p_1, \mathcal{M}_2/q_1, \mathcal{M}_3/r, \mathcal{M}_4/s, \mathcal{R}, i_{in}, i_{out}),$$

with the following components:

$$\begin{array}{rcl}
\Gamma & = & \Sigma \cup \{a_i \mid 1 \leq i \leq n\} \cup \{b_{i,j} \mid 1 \leq i \leq n, 1 \leq j \leq m+1\} \\
 & \cup & \{c_i, d_{i,0}, d_{i,1} \mid 1 \leq i \leq m\} \cup \{g_i \mid 1 \leq i \leq mn+3n+4m\} \\
 & \cup & \{a_{n+1}, d_{m+1,0}, h, \texttt{yes}, \texttt{no}\}, \\
\Sigma & = & \{x_{i,j}, \bar{x}_{i,j} \mid 1 \leq i \leq n, 1 \leq j \leq m\}, \\
P & = & \{p_i, q_i \mid 1 \leq i \leq n+1\} \cup \{\bar{p}_i \mid 2 \leq i \leq n+1\} \cup \{r, s\}, \\
\mathcal{E} & = & \{c_i, d_{i,0}, d_{i,1} \mid 1 \leq i \leq m\} \cup \{b_{i,j} \mid 1 \leq i \leq n, 1 \leq j \leq m+1\} \\
 & \cup & \{g_i \mid 1 \leq i \leq mn+3n+4m\},
\end{array}$$

$$\begin{aligned}
\mathcal{M}_1 &= \{a_1, b_{2,1}, b_{3,1}, \ldots, b_{n,1}, d_{1,0}\}, \mathcal{M}_2 = \{b_{1,1}\}, \\
\mathcal{M}_3 &= \{\texttt{yes}, \texttt{no}\}, \mathcal{M}_4 = \{g_1\}, \\
i_{in} &= 1 \text{ is the input cell}, \\
i_{out} &= 0 \text{ is the output zone},
\end{aligned}$$

The set $\mathcal{R}$ of rules consists of the following rules:

$r_{1,i} \equiv [\ p_i \mid a_i\ ]_1 \rightarrow [\ p_{i+1} \mid h\ ]_1 [\ \bar{p}_{i+1} \mid h\ ]_1$, $1 \leq i \leq n$.

$r_{2,i} \equiv [\ \bar{p}_i \mid a_i\ ]_1 \rightarrow [\ p_{i+1} \mid h\ ]_1 [\ \bar{p}_{i+1} \mid h\ ]_1$, $2 \leq i \leq n$.

$r_{3,i,j} \equiv (1, (p_{i+1}, x_{i,j})/c_j, 0)$, $1 \leq i \leq n, 1 \leq j \leq m$.

$r_{4,i,j} \equiv (1, (\bar{p}_{i+1}, \bar{x}_{i,j})/c_j, 0)$, $1 \leq i \leq n, 1 \leq j \leq m$.

$r_{5,i,j} \equiv (2, (q_i, b_{i,j})/b_{i,j+1}, 0)$, $1 \leq i \leq n, 1 \leq j \leq m$.

$r_{6,i} \equiv [\ q_i \mid b_{i,m+1}\ ]_2 \rightarrow [\ q_{i+1} \mid a_{i+1}\ ]_2 [\ q_{i+1} \mid a_{i+1}\ ]_2$,
$1 \leq i \leq n$.

$$r_{7,i} \equiv (1, (p_i, b_{i,1})/(q_i, a_i), 2),\ 2 \leq i \leq n.$$

$$r_{8,i} \equiv (1, (\bar{p}_i, b_{i,1})/(q_i, a_i), 2),\ 2 \leq i \leq n.$$

$$r_{9,i} \equiv (1, (q_i, \lambda)/(p_i, \lambda), 2),\ 2 \leq i \leq n.$$

$$r_{10,i} \equiv (1, (q_i, \lambda)/(\bar{p}_i, \lambda), 2),\ 2 \leq i \leq n.$$

$$r_{11,j} \equiv (1, (p_{n+1}, c_j d_{j,0})/(q_{n+1}, \lambda), 2),\ 1 \leq j \leq m.$$

$$r_{12,j} \equiv (1, (\bar{p}_{n+1}, c_j d_{j,0})/(q_{n+1}, \lambda), 2),\ 1 \leq j \leq m.$$

$$r_{13,j} \equiv (2, (p_{n+1}, d_{j,0})/d_{j,1}, 0),\ 1 \leq j \leq m.$$

$$r_{14,j} \equiv (2, (\bar{p}_{n+1}, d_{j,0})/d_{j,1}, 0),\ 1 \leq j \leq m.$$

$$r_{15,j} \equiv (1, (q_{n+1}, \lambda)/(p_{n+1}, d_{j,1}), 2),\ 1 \leq j \leq m.$$

$$r_{16,j} \equiv (1, (q_{n+1}, \lambda)/(\bar{p}_{n+1}, d_{j,1}), 2),\ 1 \leq j \leq m.$$

$$r_{17,j} \equiv (1, (p_{n+1}, d_{j,1})/d_{j+1,0}, 0),\ 1 \leq j \leq m.$$

$$r_{18,j} \equiv (1, (\bar{p}_{n+1}, d_{j,1})/d_{j+1,0}, 0),\ 1 \leq j \leq m.$$

$$r_{19} \equiv (1, (p_{n+1}, d_{m+1,0})/(r, \texttt{yes}), 3).$$

$$r_{20} \equiv (1, (\bar{p}_{n+1}, d_{m+1,0})/(r, \texttt{yes}), 3).$$

$$r_{21} \equiv (1, (r, \texttt{yes})/\lambda, 0).$$

$$r_{22,i} \equiv (4, (s, g_i)/g_{i+1}, 0),\ 1 \leq i \leq mn + 3n + 4m - 1.$$

$$r_{23} \equiv (4, (s, g_{mn+3n+4m})/(r, \lambda), 3).$$

$$r_{24} \equiv (3, (s, g_{mn+3n+4m}\texttt{no})/\lambda, 0).$$

the family Π is polynomially uniform by a Turing machine

- size of the set Γ: $4mn + 7m + 5n + 5 \in O(mn)$;
- size of the set P: $3n + 4 \in O(n)$;
- initial number of cells: $4 \in O(1)$;
- initial number of objects: $n + 5 \in O(n)$;
- initial number of proteins: $4 \in O(1)$;
- number of rules: $4mn + 10n + 12m - 1 \in O(mn)$;
- maximum length of a rule: $4 \in O(1)$.

the family Π is polynomially bounded

- if the formula C is satisfiable, the computation takes $mn + 3n + 4m$ steps;

the family Π is polynomially bounded

- if the formula C is satisfiable, the computation takes $mn + 3n + 4m$ steps;
- if the formula C is not satisfiable, the computation takes $mn + 3n + 4m + 1$ steps.

Open problems

- the computational efficiency of such P systems without environment;

Open problems

- the computational efficiency of such P systems without environment;
- if we consider division rules that are inspired only by proteins, then what is the computational efficiency of such P systems;

Open problems

- the computational efficiency of such P systems without environment;
- if we consider division rules that are inspired only by proteins, then what is the computational efficiency of such P systems;
- whether the length of communication rules used is optimal;

Open problems

- the computational efficiency of such P systems without environment;
- if we consider division rules that are inspired only by proteins, then what is the computational efficiency of such P systems;
- whether the length of communication rules used is optimal;
- cell separation instead of cell division.

References

1. A. Păun, B. Popa, P systems with protein on membranes. Fundamenta Informaticae 72 (2006) 467–483.
2. Gh. Păun, G. Rozenberg, A. Salomaa (Eds), Handbook of Membrane Computing, Oxford University Press, 2010.

Thank you for your attention!