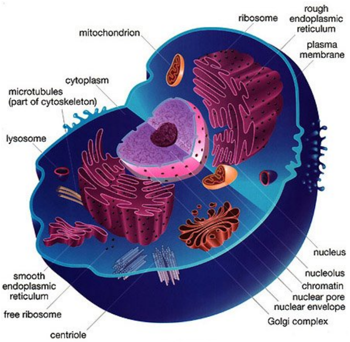
An Introduction to Membrane Computing

A. Riscos-Núñez

SCORE Lab
Research Group on Natural Computing
Universidad de Sevilla

3rd SCORE Workshop & 19th BWMC
January 24, Sevilla

Can cells compute?



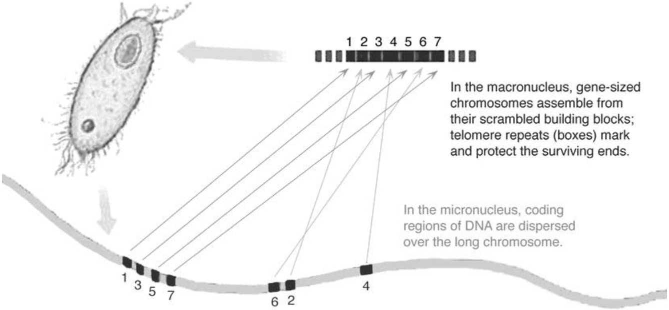
Computation is out there

Arithmetics

- They can count (threshold): *quorum sensing*
- They can distribute / divide: *mitosis*

Memory pointers

- Genes self-assembly in Cilliates



Membrane Computing

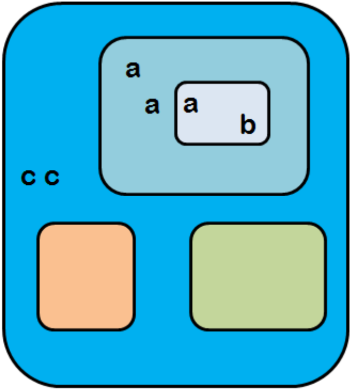


Figure: A P system

- Multisets of objects
  - distinguished alphabets
- Membranes (regions)
  - a.k.a. cells, neurons
  - input / output regions
- Rules
  - Objects
  - Membranes
- Environment

Membrane Computing

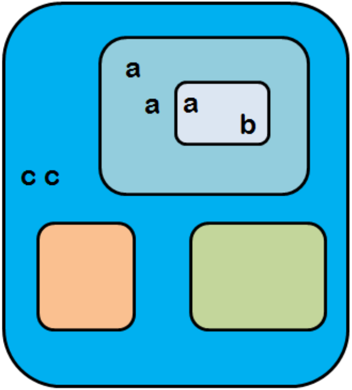


Figure: A P system

- Machine-oriented model.
- Non-deterministic devices.
  - ($\neq$ random!)
- Two levels of parallelism (objects & membranes).
- Global clock.

It has developed quickly into a **vigorous** scientific discipline.

- ⋆ International Conference on Membrane Computing (**23rd** edition).
- ⋆ Brainstorming Week on Membrane Computing (**19th** edition).
- ⋆ Asian Conference on Membrane Computing (**11th** edition).

In 2003, Thomson Institute for Scientific Information (**ISI**) declared this area as a **Fast Emerging Research Front in Computer Science**

In 2016, **International Membrane Computing Society** was founded

In 2019, 1st volume of **Journal of Membrane Computing (Springer)**

It has developed quickly into a **vigorous** scientific discipline.

- ⋆ International Conference on Membrane Computing (**23rd** edition).
- ⋆ Brainstorming Week on Membrane Computing (**19th** edition).
- ⋆ Asian Conference on Membrane Computing (**11th** edition).

In 2003, Thomson Institute for Scientific Information (**ISI**) declared this area as a **Fast Emerging Research Front in Computer Science**

In 2016, **International Membrane Computing Society** was founded

In 2019, 1st volume of **Journal of Membrane Computing (Springer)**

The *power* of a computing model

Expressive power

- Languages (Sets) that can be generated (accepted)
  - Chomsky hierarchy (REG, . . . , RE)
  - Automata theory (FA, . . . , TM)

Efficiency

- Class of problems that can be solved *efficiently*

Diversity of definitions

Syntax and Semantics

Objects

- strings, arrays, spikes, . . .

Membranes

- tree-like / graph-like structure
- labels, charges, proteins, . . .

Rules

- selecting which **types**
  (e.g. forbidding dissolution, using only communication, . . .)
- controlling **applicability**
  (e.g. priorities, permitting / forbidding conditions, alternatives to maximal parallelism, . . .)

Diversity of interpretations

- *Generative* devices: fixed initial configuration, we collect the outputs of all the non-deterministic computations.
- *Computing* devices: given an input (encoded somehow), compute the resulting output (multiset).
- *Decision* tools: special objects *yes* and *no*, s.t. their presence / absence in the output decides whether the given input was accepted by the P system or not.
- *Simulation* tools: no halting configuration, the output is the computation.

Diversity of interpretations

- *Generative* devices: fixed initial configuration, we collect the outputs of all the non-deterministic computations.
- *Computing* devices: given an input (encoded somehow), compute the resulting output (multiset).
- *Decision* tools: special objects *yes* and *no*, s.t. their presence / absence in the output decides whether the given input was accepted by the P system or not.
- *Simulation* tools: no halting configuration, the output is the computation.

Diversity of interpretations

- *Generative* devices: fixed initial configuration, we collect the outputs of all the non-deterministic computations.
- *Computing* devices: given an input (encoded somehow), compute the resulting output (multiset).
- *Decision* tools: special objects *yes* and *no*, s.t. their presence / absence in the output decides whether the given input was accepted by the P system or not.
- *Simulation* tools: no halting configuration, the output is the computation.

Diversity of interpretations

- *Generative* devices: fixed initial configuration, we collect the outputs of all the non-deterministic computations.
- *Computing* devices: given an input (encoded somehow), compute the resulting output (multiset).
- *Decision* tools: special objects *yes* and *no*, s.t. their presence / absence in the output decides whether the given input was accepted by the P system or not.
- *Simulation* tools: no halting configuration, the output is the computation.

Main research directions

- Theoretical Foundations
  - **Universality** results
    Generative / accepting power equivalent to . . .
  - What if . . . ?
  - Formalization
- Computational Complexity
  - **Efficient** solutions to **hard** problems
  - **P conjecture**
- Practical Approach
  - **Simulators**
  - **Modelling**
  - Generative music, Robot control, Model checking, . . .

Based on a hierarchical arrangement of membranes delimiting **compartments** where multisets of chemicals **evolve** according to given evolution rules.

- The rules are either modeling chemical reactions (in the form of multiset rewriting rules), or they are inspired by other biological processes (passing objects through membranes, mitosis, etc.) and have the form of communication rules, division rules, etc.

Initial models: Cell-like approach

- $\Pi = (\Gamma, \Sigma, H, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{in}, i_{out})$.
- *Basic transition* P systems:
  - $[\ u\ ]_h \rightarrow [\ v\ ]_h$ (evolution rules).
  - $[\ u\ ]_h \rightarrow v\ [\ \ ]_h$ and $u\ [\ \ ]_h \rightarrow [\ v\ ]_h$ (communication rules).
  - $[\ u\ ]_h \rightarrow v$ (dissolution rules).
- $\mathcal{T}$: class of recognizer basic transition P systems.

Initial models: Cell-like approach

- $\Pi = (\Gamma, \Sigma, H, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{in}, i_{out})$.
- *Basic transition* P systems:
  - $[\ u\ ]_h \rightarrow [\ v\ ]_h$ (evolution rules).
  - $[\ u\ ]_h \rightarrow v\ [\ \ ]_h$ and $u\ [\ \ ]_h \rightarrow [\ v\ ]_h$ (communication rules).
  - $[\ u\ ]_h \rightarrow v$ (dissolution rules).
- $\mathcal{T}$: class of recognizer basic transition P systems.

P systems with active membranes

(a) $[\, a \rightarrow u \,]_h^{\alpha}$ (*object evolution* rules).

(b) $a\,[\;\;]_h^{\alpha_1} \rightarrow [\, b \,]_h^{\alpha_2}$ (*send–in communication* rules).

(c) $[\, a \,]_h^{\alpha_1} \rightarrow [\;\;]_h^{\alpha_2}\, b$ (*send–out communication* rules).

(d) $[\, a \,]_h^{\alpha} \rightarrow b$ (*dissolution* rules).

(e) $[\, a \,]_h^{\alpha_1} \rightarrow [\, b \,]_h^{\alpha_2}\, [\, c \,]_h^{\alpha_3}$ (*division* rules for *elementary membranes*).

(f) $[\, [\,]_{h_1}^{\alpha_1}\, [\,]_{h_2}^{\alpha_2} \,]_h^{\alpha} \rightarrow [\, [\,]_{h_1}^{\alpha_3} \,]_h^{\beta}\, [\, [\,]_{h_2}^{\alpha_4} \,]_h^{\gamma}$ (*division* rules for *non–elementary membranes*).

P Systems with Active Membranes (cont.)

1. Non deterministic, maximally parallel evolution.
2. For each membrane, only one from b), c), d), e) and f).
3. Dissolution $\Rightarrow$ transfer to the parent.
4. Division $\Rightarrow$ duplicate and transfer to the new membranes.

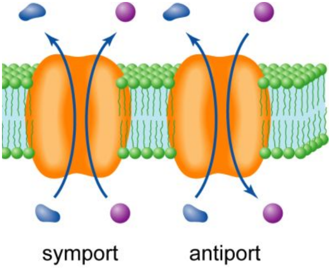
Three* electrical charges, without cooperation and without priorities.

Inspired by

- intercellular communication
- cooperation between neurons

- Communication rules: *symport/antiport*
- Cells as nodes of a graph (and environment)

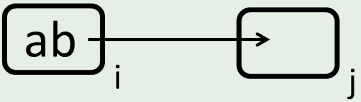
Tissue-like P systems



Based on the complex communication networks established among adjacent cells by making their protein channels cooperate, moving molecules directly from one cell to another.

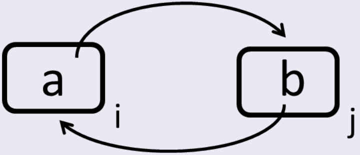
Symport/antiport rules

Example (symport)



$$(i, ab/\lambda, j)$$

Example (antiport)



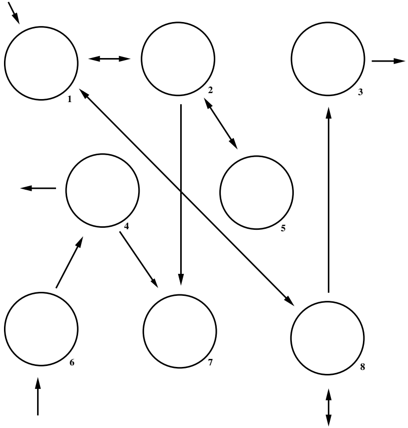
$$(i, a/b, j)$$

Tissue-like P systems

Symport/antiport rules define a directed graph in an **implicit** way.

Rules (i, u/v , j)

$(0, ba^2/\lambda, 1), (0, \lambda/b^4cd, 3),$
$(0, \lambda/ab^3, 4), (0, c/\lambda, 6),$
$(0, a/b^2, 8), (1, c^3/b^2, 2)$
$(1, ad/a, 8), (2, ab/\lambda, 7),$
$(2, b/b^2, 5), (3, \lambda, d^2, 8),$
$(4, \lambda/a, 6), (4, b^2c^2/\lambda, 7)$



Finite alphabets

- working alphabet (Γ),
- **input** alphabet ($\Sigma \subseteq \Gamma$),
- **environment** alphabet ($\mathcal{E} \subseteq \Gamma \setminus \Sigma$)

Rules

- symport/antiport: $(i, u/v, j)$
- cell division: $[a]_i \rightarrow [b]_i[c]_i$
- cell separation: $[\ a\ ]_i \rightarrow [\ \Gamma_1\ ]_i[\ \Gamma_2\ ]_i$ (for a fixed partition)

Length of rule $(i, u/v, j)$ = $|u| + |v|$

Tissue-like P systems with Cell Division / Separation

Initial configuration

- multisets $\mathcal{M}_1, \dots, \mathcal{M}_q$ over $\Gamma \setminus \Sigma$
- environment

- Non deterministic, maximally parallel.
- While dividing / separating, communication is blocked.
- Division $\Rightarrow$ duplicate and transfer to the new cells.
- Separation $\Rightarrow$ objects are distributed among the new cells.

Cell Division: $[a]_i \rightarrow [b]_i[c]_i$

Contents duplicated

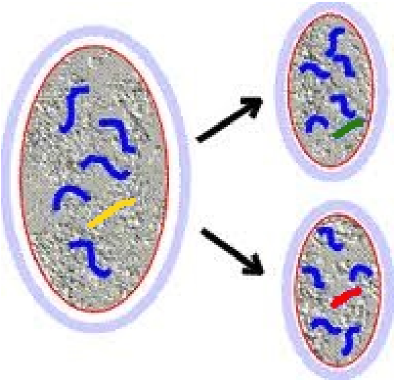
$a^5 \quad b \quad c^2 \quad d^3$

Cell Division: $[a]_i \rightarrow [b]_i[c]_i$

Contents duplicated

$a^4 \; b^2 \; c^2 \; d^3$

$a^4 \; b \; c^3 \; d^3$



Cell Separation: $[a]_i \to [\Gamma_1]_i[\Gamma_2]_i$

Contents distributed

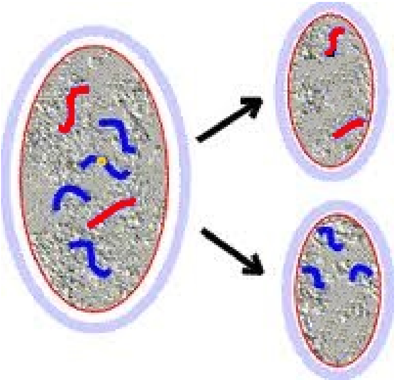
$a^5 \; b \; c^2 \; d^3$

Cell Separation: $[a]_i \to [\Gamma_1]_i[\Gamma_2]_i$

Contents distributed

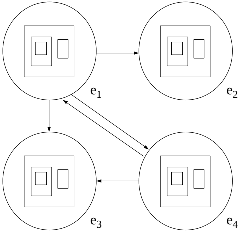
$a^4 \quad c^2$

$b \quad d^3$



Multienvironment P systems

Population Dynamics P systems (probabilistic)



Skeleton rules

$$u\,[\,v\,]_h^{\alpha} \xrightarrow{f_r} u'\,[\,v'\,]_h^{\beta}$$

Environment rules

$$(a)_{e_j} \xrightarrow{f_r} (b)_{e_k}$$

Algorithms for probabilistic behaviour

- Binomial Block Based (**BBB**) simulation algorithm
- Direct Non-Deterministic distribution algorithm with Probabilities (**DNDP**)
- Direct distribution based on Consistent Blocks Algorithm (**DCBA**)
- ...

Neural-like approach

Inspiration

Informally, an SN P system consists of a set of **neurons** placed in the nodes of a **directed graph** which send signals (called *spikes*) along the arcs of the graph (representing *synapses* between neurons).

Neural-like approach

- *Spiking rules* and *Forgetting rules*
- Applicability w.r.t. all spikes present in a neuron (although maybe not all of them consumed). Selects **one** rule
- Produced spikes are sent via all outgoing synapses (maybe with a delay)
- Output neuron spikes into the environment

Neural-like approach

- *Spiking rules* and *Forgetting rules*
- Applicability w.r.t. all spikes present in a neuron (although maybe not all of them consumed). Selects **one** rule
- Produced spikes are sent via all outgoing synapses (maybe with a delay)
- Output neuron spikes into the environment

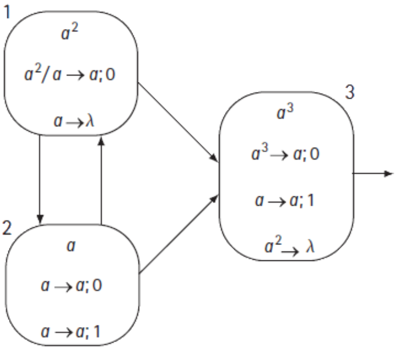
Neural-like approach

- *Spiking rules* and *Forgetting rules*
- Applicability w.r.t. all spikes present in a neuron (although maybe not all of them consumed).
  Selects **one** rule
- Produced spikes are sent via all outgoing synapses (maybe with a delay)
- Output neuron spikes into the environment

Neural-like approach

- *Spiking rules* and *Forgetting rules*
- Applicability w.r.t. all spikes present in a neuron (although maybe not all of them consumed). Selects **one** rule
- Produced spikes are sent via all outgoing synapses (maybe with a delay)
- Output neuron spikes into the environment

An example



An SN P system generating all natural numbers greater than 1.

And many more . . .

Hot topic!

- weigths, astrocytes, . . .
- autapses, structural plasticity, . . .
- fuzzy values (FRSNP systems)
- . . .

Thanks for your attention!

Questions and feedback welcome!