# P Systems and Membrane Computation

Leroy Todd
COSC 594 Fall 2018

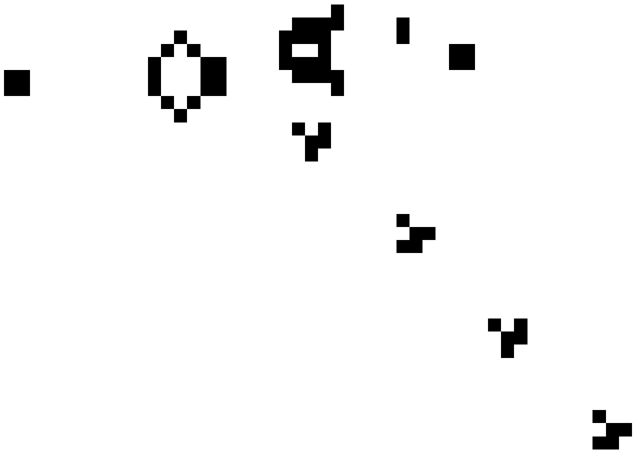
THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

# It is worth learning from biology

- Aldeman and DNA Computation showed real evidence for biological computation
- Can we glean anything more computationally significant?
- How can we abstract computation from the ways cells operate?

# Cells are not just things to be processed

- It is very easy to mimic life rather than recreate its effect
- ANN Neurons have no inner structure, can be thought of as simple functions
- Cellular automata are another example of a system where cells are just state



File:Gospers glider gun.gif. (2017, April 18). Wikimedia Commons, the free media repository. Retrieved 00:50, November 29, 2018 from https://commons.wikimedia.org/w/index.php?title=File:Gospers_glider_gun.gif&oldid=241299510.

# Computation is not structure alone

- Unlike other models, e.g. ANN, the structure of cells alone does not determine its output
- Cellular computation must rely on some sort of action from the cell itself
- We want to explore cell organization

# P Systems

- Finalized in 1998 by Gheorghe Păun
- A system that abstracts computation from cell structure to produce cells that organize
- A formalization of membrane computation

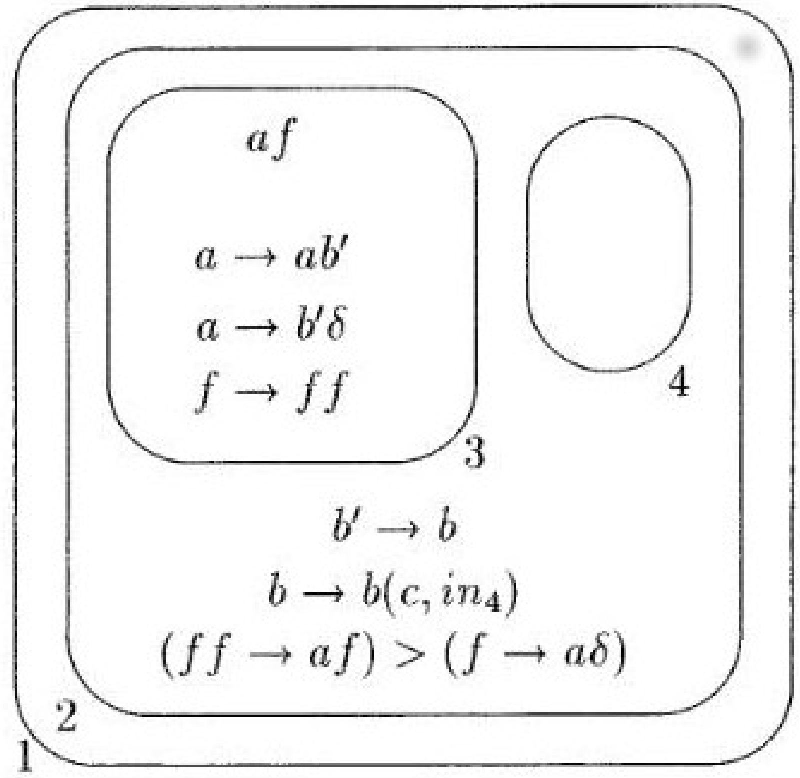



Figure 1: Martín-Vide C, et al. Machines, Comptutations, and Universality (2001)

# P Systems have several classes

- Transition
- Symport/Antiport
- Communication
- Evolution
- Active Membrane

# Structure for all classes of P Systems

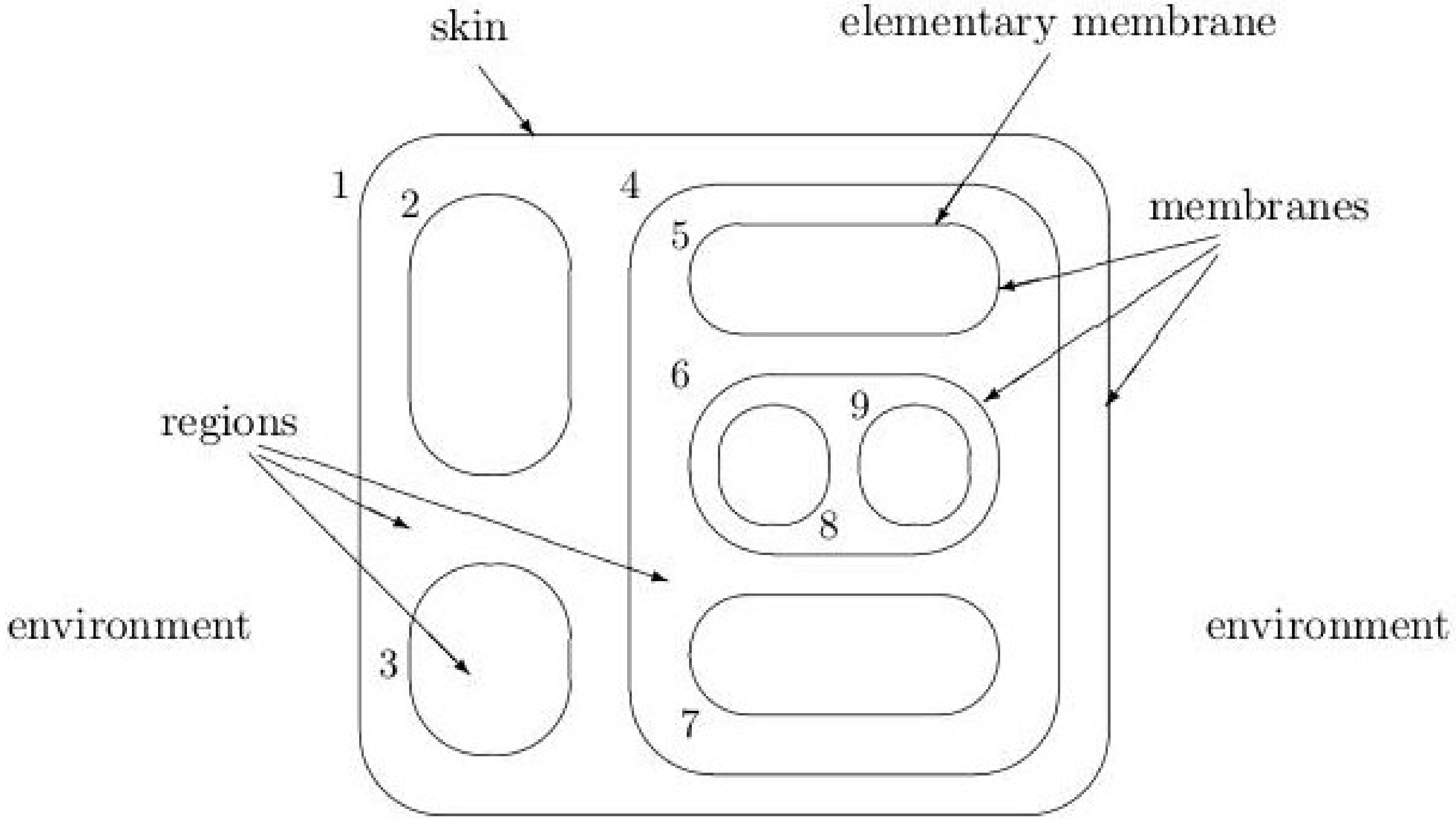



Figure 2: Păun G. Applications of Membrane Computing (2006)

# Symbol Structure

- Each membrane has a unique label
- Entire system can be represented as a tree
- Can also be represented as a string of labeled and nested brackets

$$[_1 \, [_2 \, ]_2 \, [_3 \, ]_3 \, [_4 \, [_5 \, ]_5 \, [_6 \, [_8 \, ]_8 \, [_9 \, ]_9 \, ]_6 \, [_7 \, ]_7 \, ]_4 \, ]_1 .$$

Figure 3: Păun G. Applications of Membrane Computing (2006)

# Membranes contain chemicals

- Chemicals are represented by symbol objects

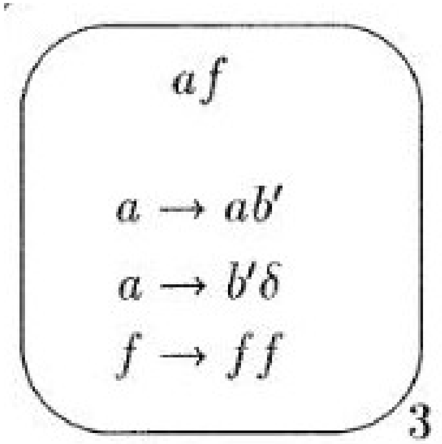



Figure 4: Păun G. Applications of Membrane Computing (2006)

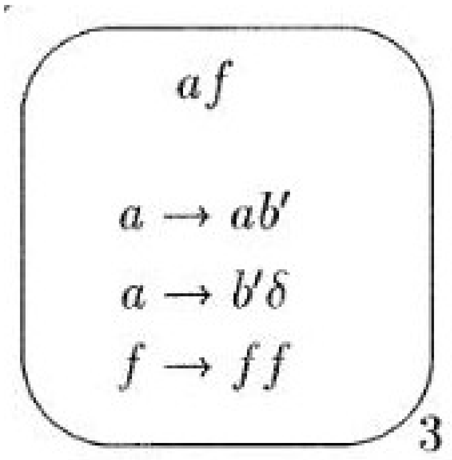
- Symbols can also represent the multiset of objects, e.g. $a^2b^3c^2$

# Rules

- Computation is in part done via rules
- Each class of P System has a different rule system (small but fundamental differences)
- Rules operate over the objects in a membrane

# Evolution Rule

- The most common type of rule (most applicable)
- It is a rewriting rule where the input is replaced with the output
- In the form *input* → *output* where *input* and *output* both represent a multiset of objects

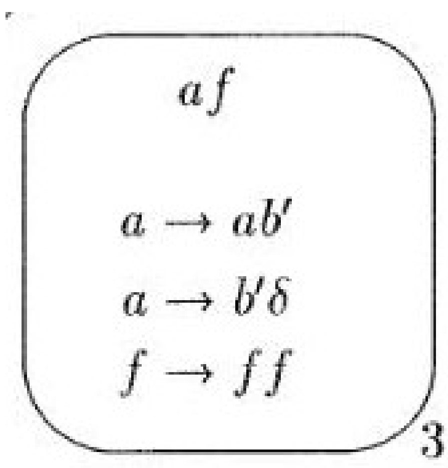


# Dissolve Symbol

- Any rule may have symbol δ which means the membrane dissolves after applying that rule
- Dissolved membranes release any objects they contain to the parent membrane
- The rules of the dissolved membranes are destroyed
- The skin cannot dissolve

# Rule Application

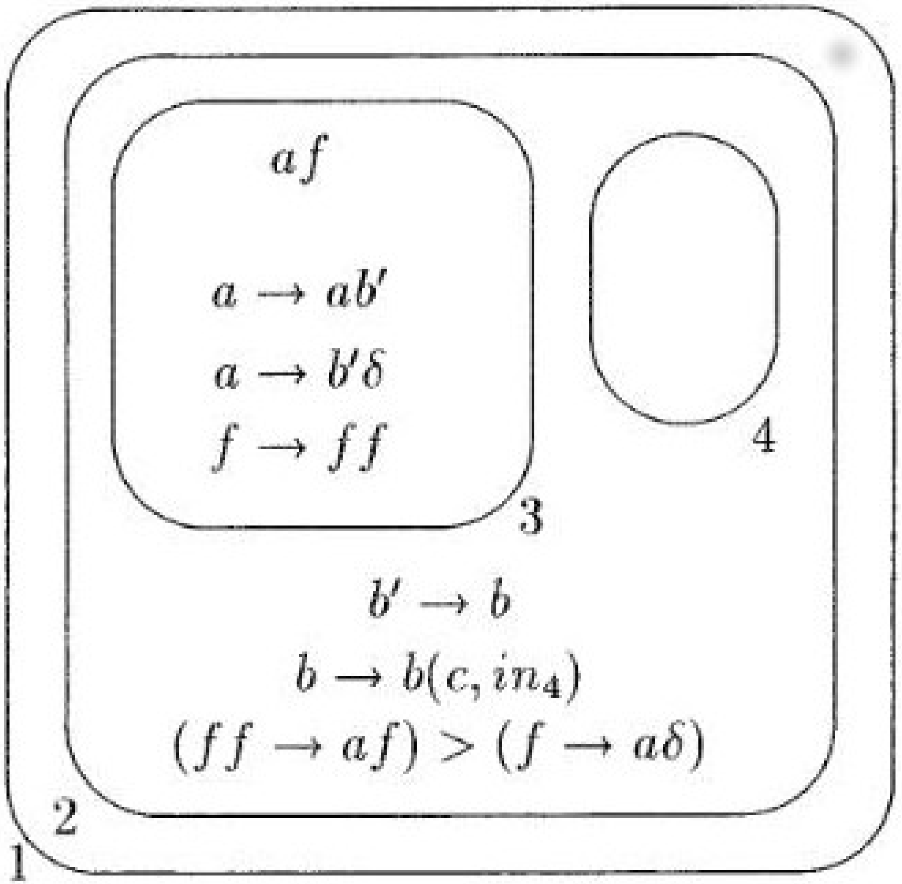
- Rules must be maximally applied
  - All rules that can be applied must be applied
- Rules that can be applied are done in a non-deterministic order for some time step
- Rules can be prioritized using >
  - e.g. ab → a > b → δ



# Target Indicators

- Rules and rule objects can have target indicators
- *Here:* object stays in membrane
- *Out:* object enters the parent membrane
- *In:* object moves to child membrane
- If an *out* indicator is given on the skin membrane, then that object enters the environment

# Computing $N^2$ Numbers



# Halting and Output

- Just like Turing Machines, need a halting configuration
- Can theoretically have infinite loops
- Typically a finite amount of objects (chemicals) sets an upper limit
- Typically halt by running out of objects
- Output determined usually by counting objects in some location

# Active Membrane Computation

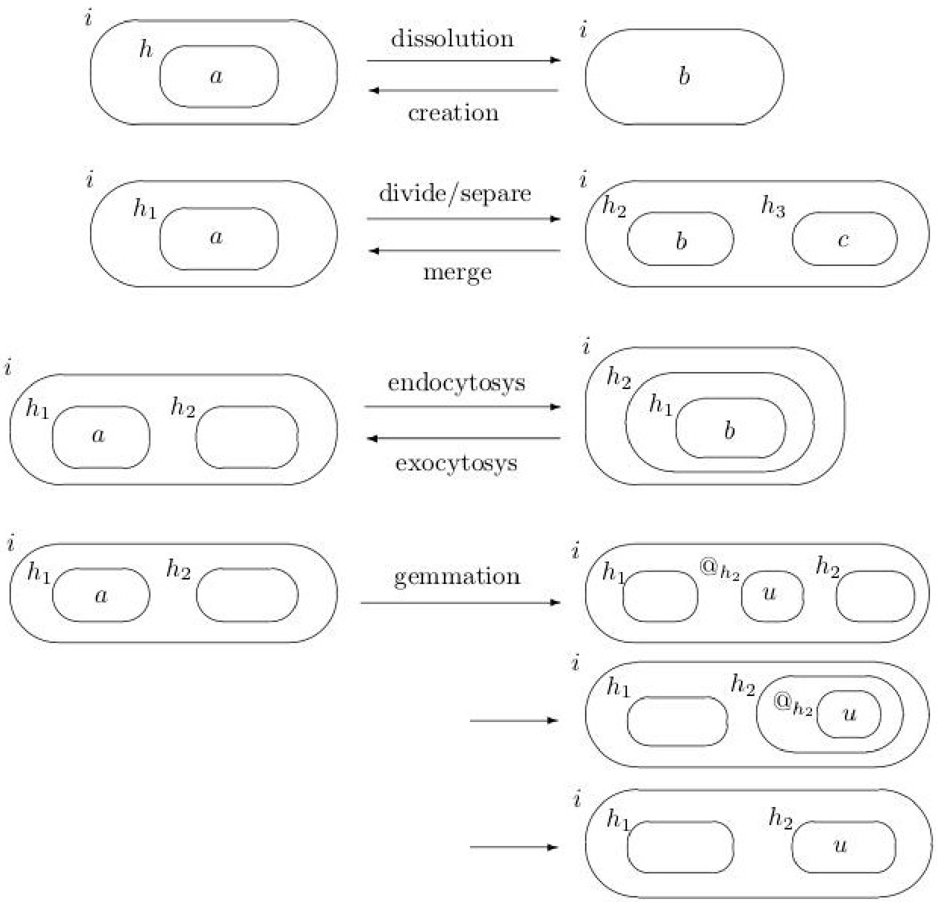



Figure 5: Păun G. Applications of Membrane Computing (2006)

# Evolution Rule

(a) $[_h a \to v]_h^e$,
for $h \in H, e \in \{+, -, 0\}, a \in O, v \in O^*$
(object evolution rules, associated with membranes and depending on the label and the charge of the membranes, but not directly involving the membranes, in the sense that the membranes are neither taking part in the application of these rules nor are they modified by them);

# *in* Communication Rule

(b) $a[_h\ ]_h^{e_1} \rightarrow [_h b]_h^{e_2}$,
for $h \in H, e_1, e_2 \in \{+, -, 0\}, a, b \in O$
(*in* communication rules; an object is introduced in the membrane, possibly modified during this process; also the polarization of the membrane can be modified, but not its label);

# *out* Communication Rule

(c) $[_h a\ ]_h^{e_1} \rightarrow [_h]_h^{e_2} b$,
for $h \in H, e_1, e_2 \in \{+, -, 0\}, a, b \in O$
(*out* communication rules; an object is sent out of the membrane, possibly modified during this process; also the polarization of the membrane can be modified, but not its label);

# Dissolving Rule

(d) $[_h a\ ]_h^e \rightarrow b$,
for $h \in H, e \in \{+, -, 0\}, a, b \in O$
(dissolving rules; in reaction with an object, a membrane can be dissolved, while the object specified in the rule can be modified);

# Division Rule

(e) $[_h a ]_h^{e_1} \rightarrow [_h b ]_h^{e_2} [_h c ]_h^{e_3}$,
for $h \in H, e_1, e_2, e_3 \in \{+, -, 0\}, a, b, c \in O$
(division rules for elementary membranes; in reaction with an object, the membrane is divided into two membranes with the same label, possibly of different polarizations; the object specified in the rule is replaced in the two new membranes by possibly new objects; the remaining objects are duplicated and may evolve in the same step by rules of type $(a)$).

# Solving SAT in Linear Time

- Runs in $2n + 2m + 1$ steps
- $(x_1 \mid x_2) \,\&\, (\sim x_1 \mid \sim x_2)$
- Creates $2^n$ variable assignment sets
- Creates *m* clause structures for each set
- Calling each of these “substructures”
- Produces *t* object if satisfiable on final step

# Creating $2^n$ Substructures

$[_0 c_i \to c_{i+1}]_0^\alpha$, for all $0 \le i \le 2n+m-2$ and $\alpha \in \{+,-,0\}$

(we count to $2n+m-1$, which is the time needed for producing all $2^n$ truth-assignments for the $n$ variables, as well as $2^n$ membrane sub-structures which will examine the truth value of formula $\gamma$ for each of these truth-assignments; this counting is done in the central membrane, irrespective which is its polarity);

$[_0 c_{2n+m-1}]_0^0 \to t$

(after $2n+m-1$ steps, each copy of membrane 0 is dissolved and their contents is released in the upper membranes, those labeled with 1);

# Creating $2^n$ Substructures (cont.)

$$[_0 a_i]_0^0 \to [_0 t_i]_0^+ [_0 f_i]_0^- \text{, for all } 1 \le i \le n$$

(in membrane 0, when it is "electrically neutral", we non-deterministically choose one variable $x_i$ and both values *true* and *false* are associated with it, in the form of objects $t_i, f_i$, which are separated in two membranes with the label 0 which differ only by these objects $t_i, f_i$ and by their charge);

$$[_{i+1}[_i\ ]_i^+[_i\ ]_i^-]_{i+1}^0 \to [_{i+1}[_i\ ]_i^0]_{i+1}^+[_{i+1}[_i\ ]_i^0]_{i+1}^-\text{, for all } 0 \le i \le m-2\text{, and}$$

$$[_m[_{m-1}\ ]_{m-1}^+[_{m-1}\ ]_{m-1}^-]_m^0 \to [_m[_{m-1}\ ]_{m-1}^0]_m^0[_m[_{m-1}\ ]_{m-1}^0]_m^0$$

(division rules for membranes labeled with $0, 1, \ldots, m$; the opposite polarization introduced when dividing a membrane 0 is propagated from lower levels to upper levels of the membrane structure and the membranes are continuously divided until dividing also membrane $m$ – which will get neutral charge).

# Substructure Satisfiability

$[_j t_i]_j^0 \rightarrow t_i$, if $x_i$ appears in clause $C_j$, $1 \leq i \leq n, 1 \leq j \leq m$, and

$[_j f_i]_j^0 \rightarrow f_i$, if $\sim x_i$ appears in clause $C_j$, $1 \leq i \leq n, 1 \leq j \leq m$

(a membrane with label $j$, $1 \leq j \leq m$, is dissolved if and only if clause $C_j$ is satisfied by the current truth-assignment; if this is the case, then the truth values associated with the variables are released in the upper membrane, that associated with the next clause, $C_{j+1}$, otherwise these truth values remain blocked in membrane $j$ and never used at the next steps by the membranes placed above; note that, as we will see immediately, after $2n + m - 1$ steps we have $2^n$ membrane sub-structures of the form $[_m [_{m-1} \ldots [_1 \ ]_1^0 \ldots ]_{m-1}^0 ]_m^0$ working in parallel in the skin membrane);

$[_{m+1} t]_{m+1}^0 \rightarrow [_{m+1} \ ]_{m+1}^+ t$

(together with the truth-assignments, we also have the object $t$, which can be passed from a level to the upper one only by dissolving membranes; this object reaches the skin membrane if only if all membranes in a sub-structure of the form $[_m [_{m-1} \ldots [_1 \ ]_1^0 \ldots ]_{m-1}^0 ]_m^0$ are dissolved, which means that the associated truth-assignment has satisfied all the clauses, that is, the formula is satisfiable; therefore, $t$ leaves the system if and only if the formula is satisfiable; when this rule is applied, the skin membrane gets a "positive charge", so the rule can be applied only once);

# Substructure Creation

Step 0: $[_3[_2[_1[_0 c_0 a_1 a_2]_0^0]_1^0]_2^0]_3^0$;

Step 1: $[_3[_2[_1[_0 c_1 t_1 a_2]_0^+ [_0 c_1 f_1 a_2]_0^- ]_1^0]_2^0]_3^0$

Step 2: $[_3[_2[_1[_0 c_2 t_1 a_2]_0^0]_1^+ [_1[_0 c_2 f_1 a_2]_0^0]_1^- ]_2^0]_3^0$

Step 3: $[_3[_2[_1[_0 c_3 t_1 t_2]_0^+ [_0 c_3 t_1 f_2]_0^- ]_1^0]_2^0 [_2[_1[_0 c_3 f_1 t_2]_0^+ [_0 c_3 f_1 f_2]_0^- ]_1^0]_2^0]_3^0$

Step 4: $[_3[_2[_1[_0 c_4 t_1 t_2]_0^0]_1^+ [_1[_0 c_4 t_1 f_2]_0^0]_1^- ]_2^0 [_2[_1[_0 c_4 f_1 t_2]_0^0]_1^+ [_1[_0 c_4 f_1 f_2]_0^0]_1^- ]_2^0]_3^0$

Step 5: $[_3[_2[_1[_0 c_5 t_1 t_2]_0^0]_1^0]_2^0 [_2[_1[_0 c_5 t_1 f_2]_0^0]_1^0]_2^0 [_2[_1[_0 c_5 f_1 t_2]_0^0]_1^0]_2^0 [_2[_1[_0 c_5 f_1 f_2]_0^0]_1^0]_2^0]_3^0$

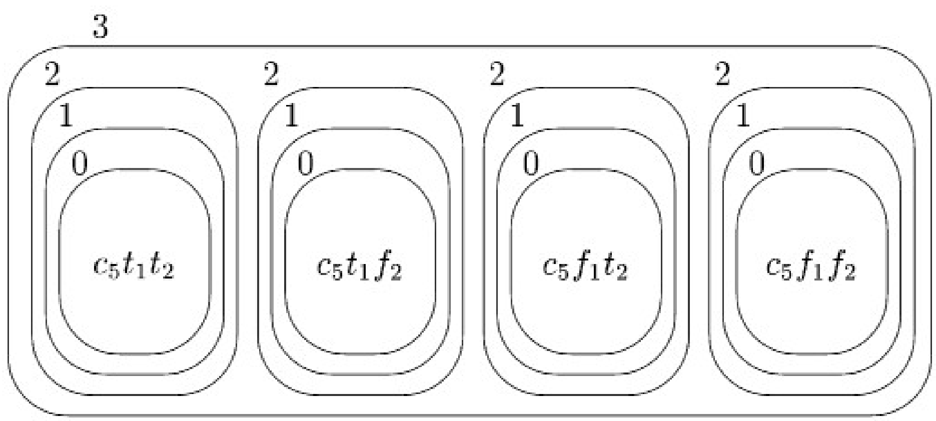



Figure 6: Păun G. P systems with active membranes: attacking NP-complete problems (2001)

# Satisfiability Tests and Halting

Step 6: $[_3[_2[_1 tt_1t_2]_1^0]_2^0[_2[_1 tt_1f_2]_1^0]_2^0[_2[_1 tf_1t_2]_1^0]_2^0[_2[_1 tf_1f_2]_1^0]_2^0]_3^0$

Step 7: $[_3[_2 tt_1t_2]_2^0[_2 tt_1f_2]_2^0[_2 tf_1t_2]_2^0[_2[_1 tf_1f_2]_1^0]_2^0]_3^0$

Step 8: $[_3[_2 tt_1t_2]_2^0 tt_1f_2tf_1t_2[_2[_1 tf_1f_2]_1^0]_2^0]_3^0$

# Current Significance

- Structure of the exponential
- Concise steps and states
- Can solve practical problems
- Still very theoretical

# Future of P Systems

- Building Physical Neural Networks that can destroy/create neurons/synapses
- Great Model for Evolutionary Computing
- Modeling of Ecosystems/Dynamical Systems
- Physical P Systems

# Electric Power System Fault Diagnosis

- Uses typical electric components
- Modeled by P Systems
- Usually a form of a Neural P System

# References


- Păun G. (2006) Introduction to Membrane Computing. In: Ciobanu G., Păun G., Pérez-Jiménez M.J. (eds) Applications of Membrane Computing. Natural Computing Series. Springer, Berlin, Heidelberg
- Martín-Vide C., Păun G. (2001) Computing with Membranes (P Systems): Universality Results. In: Margenstern M., Rogozhin Y. (eds) Machines, Computations, and Universality. MCU 2001. Lecture Notes in Computer Science, vol 2055. Springer, Berlin, Heidelberg
- Păun G. (2000) Computing with Membranes. In: Journal of Computer and System Sciences vol 61, issue 1 (August 2000), 108-143. Academic Press, Inc., Orlando, USA
- Păun G. (2001) P systems with active membranes: attacking NP-complete problems. J. Autom. Lang. Comb. 6, 1 (January 2001), 75-90.
- Zhang, G., Pérez-Jiménez, M. J., & Gheorghe, M. (2017). Real-life Applications with Membrane Computing. S.L.: Springer-Verlag.