

Faktorizáció és kapcsolódó problémák membrán rendszerekben II.

Vaszil György

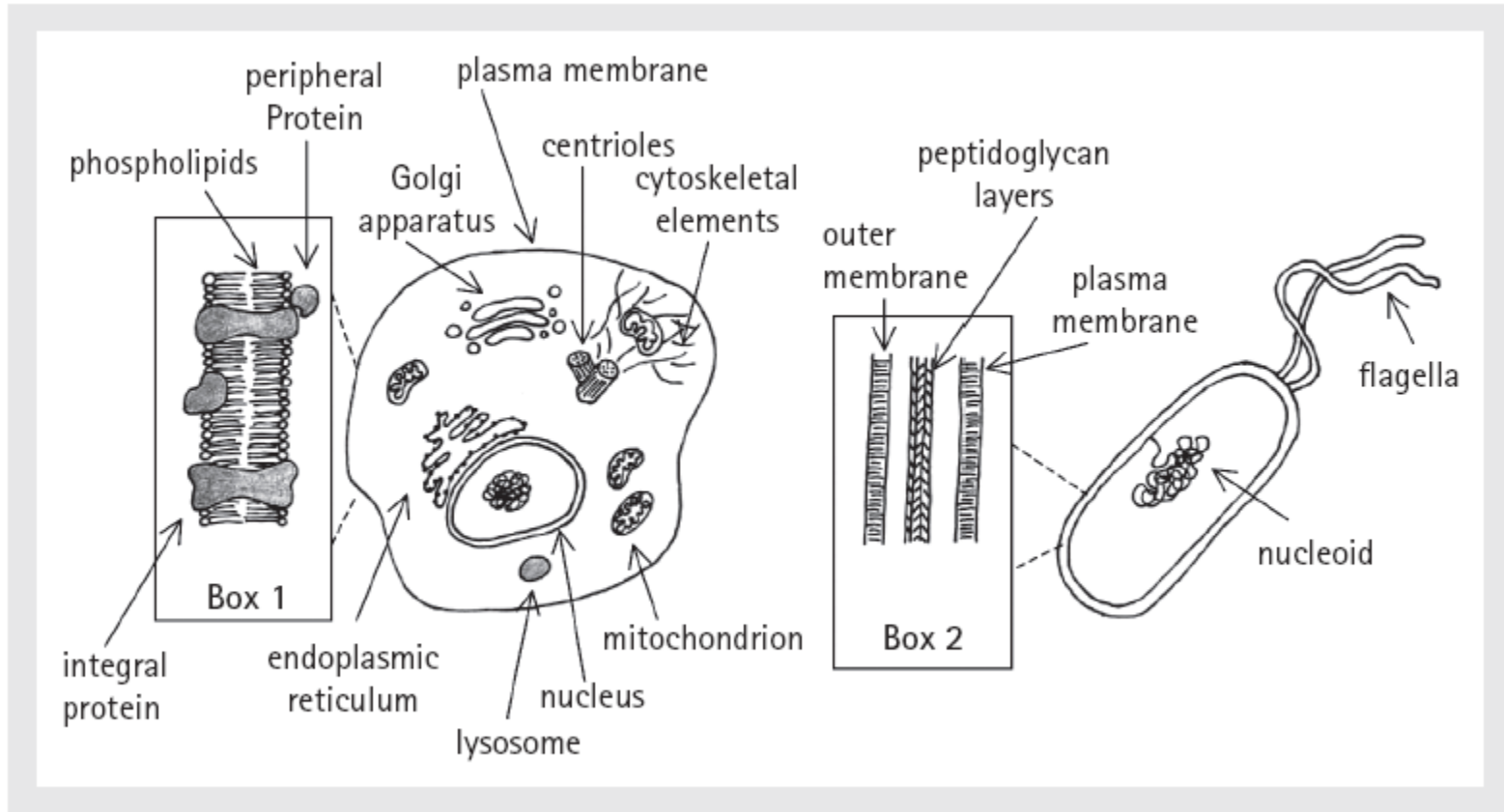
Elméleti Számítástudományi
Kutatócsoport - MTA SZTAKI

2011.1.12.

In this talk

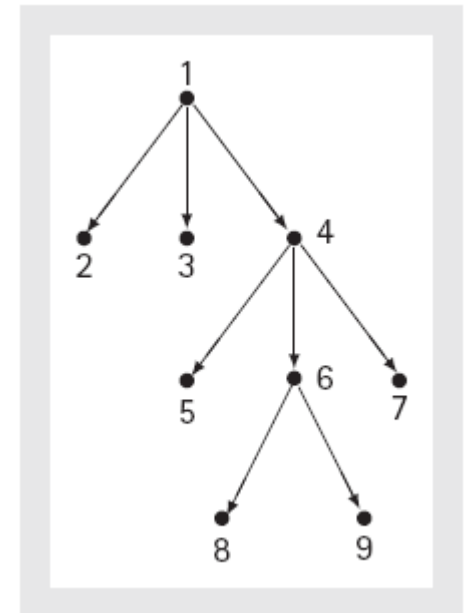
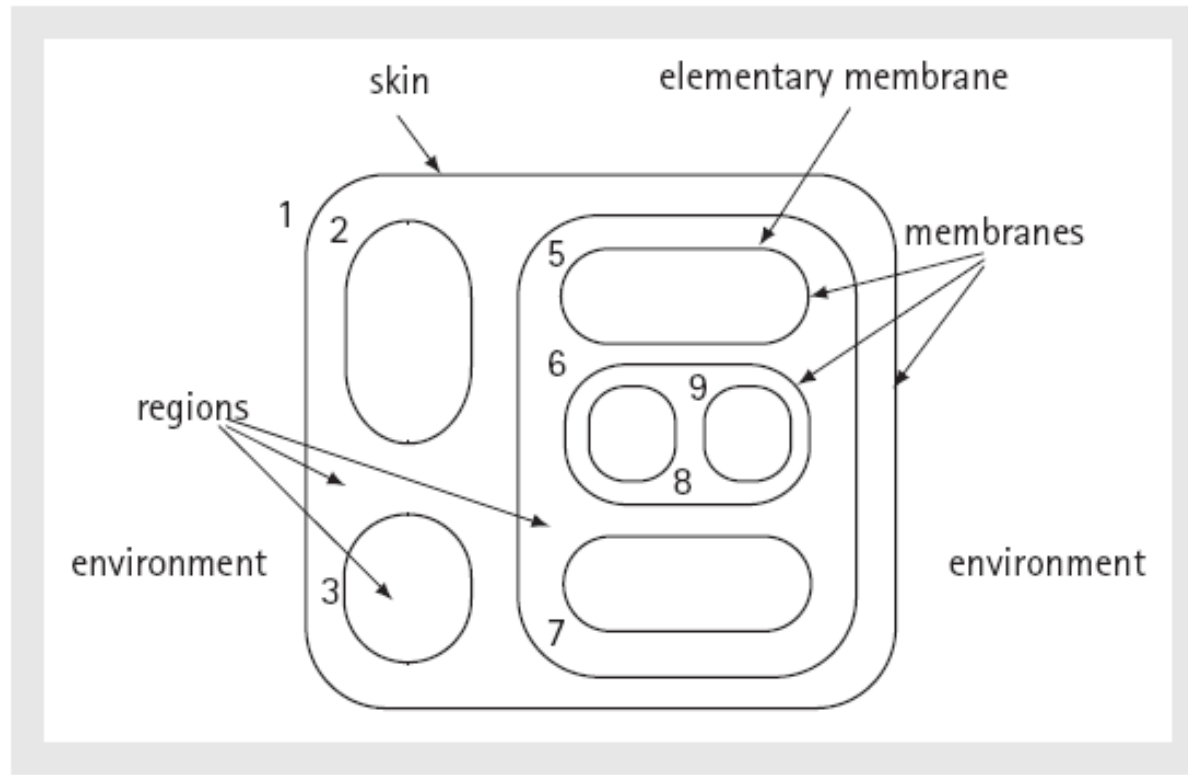
- Membrane systems (P systems) with active membranes
 - Overview, different variants, uniform and semi-uniform solutions to NP complete problems
- Discrete logarithm with membrane systems
 - The membrane system variant used for the solution, the idea of the solution

Membrane systems (P systems)



P systems, a membrane structure

A hierarchical arrangement of regions where multisets of objects evolve according to given evolutionary rules



P systems, multiset rewriting rules

The rules

- change the objects
- move the objects between neighboring regions
- manipulate the membrane structure

The rules are applied

- in parallel
- in a synchronized manner

P systems with active membranes, the rules

a) Object evolution

$$[a \rightarrow v]_h^e, \text{ for } h \in H, e \in \{+, -, 0\}, a \in O, v \in O^*$$

b) c) Communication

$$a []_h^{e_1} \rightarrow [b]_h^{e_2}, \text{ for } h \in H, e_1, e_2 \in \{+, -, 0\}, a, b \in O$$

$$[a]_h^{e_1} \rightarrow []_h^{e_2} b, \text{ for } h \in H, e_1, e_2 \in \{+, -, 0\}, a, b \in O$$

d) Membrane dissolution

$$[a]_h^e \rightarrow b, \text{ for } h \in H - \{s\}, e \in \{+, -, 0\}, a, b \in O$$

e) Membrane division

$$[a]_h^{e_1} \rightarrow [b]_h^{e_2} [c]_h^{e_3}, \text{ for } h \in H - \{s\}, e_1, e_2, e_3 \in \{+, -, 0\}, a, b, c \in O$$

The rules are applied in parallel

1. The rules are applied
 - to all objects/membranes to which they can be applied
 - all objects/membranes are used by one rule
 - any object/membrane which can evolve, should evolve
2. Objects of a dissolved membrane are left free in the parent region
3. Objects of a divided membrane are duplicated

Computations of a membrane system

- The system starts in an **initial configuration**, and
- **evolves** according to its rules,
- **by changing, creating, deleting, and moving the objects** between the regions,
- **and/or dynamically changing the membrane structure.**
- Some of the **evolutions/computations** are defined to be **successful** (no rule is applicable, a final configuration is reached, etc.), and
- these yield a **result** (a vector of multiplicities of objects in the regions)

Computational power

$$NOP_{3,3}(act_2, (a), (b), (c)) = NRE.$$

$$NOP_{2,2}(act_3, (a), (b), (c)) = NRE.$$

$$NOP_{5,*}(act_1, (a_1), (b_1), (c_1), (d_1), (e_1)) = NRE.$$

Division of non-elementary membranes

f) Membrane division II.

$$\begin{aligned}
 & [h_0 [h_1]_{h_1}^{\alpha_1} \cdots [h_k]_{h_k}^{\alpha_1} [h_{k+1}]_{h_{k+1}}^{\alpha_2} \cdots [h_n]_{h_n}^{\alpha_2}]_{h_0}^{\alpha_0} \\
 & \rightarrow [h_0 [h_1]_{h_1}^{\alpha_3} \cdots [h_k]_{h_k}^{\alpha_3}]_{h_0}^{\alpha_5} [h_0 [h_{k+1}]_{h_{k+1}}^{\alpha_4} \cdots [h_n]_{h_n}^{\alpha_4}]_{h_0}^{\alpha_6},
 \end{aligned}$$

for $k \geq 1, n > k, h_i \in H, 0 \leq i \leq n, \alpha_0, \dots, \alpha_6 \in \{+, -, 0\}$
 $\{\alpha_1, \alpha_2\} = \{+, -\}$

Difficult problems can be attacked

For example:

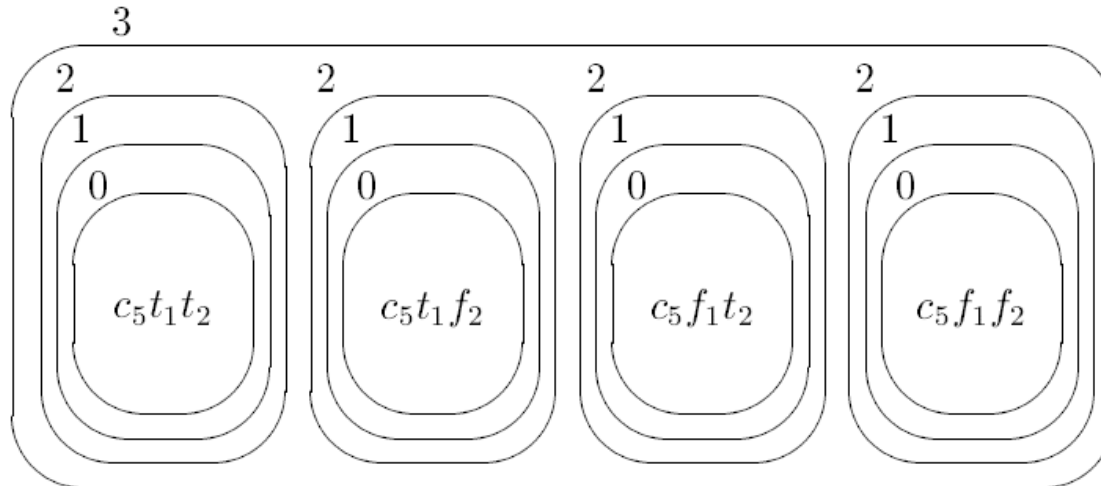
SAT can be solved by a P system in time which is linear in the number of variables and the number of clauses.

[Gh. Paun, 1999]

The satisfiability problem $(x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$

1. Start with: $[_3[_2[_1[_0c_0a_1a_2]_0^0]_1^0]_2^0]_3^0$

2. Generate all truth assignments:



3. Dissolve membrane i if an assignment satisfies the i -th clause

The satisfiability problem

There also exist “uniform” solutions:

The instance of the problem is not part of the description of the associated P system - All instances of a given size are processed by the same system with the appropriate input.

[M.A. Gutierrez-Naranjo, M.J. Perez-Jimenez, F.J. Romero campero, 2007]

Without membrane division

$$PsMAT \subseteq PsLPA(ndiv)$$

$$LPA(ndiv) - MAT \neq \emptyset$$

[Gh. Paun, 1999]

Computing the discrete logarithm

978-0-7695-3610-1/09 \$25.00 © 2009 IEEE
DOI 10.1109/NSWCTC.2009.269

438



2009 International Conference on Networks Security, Wireless Communications and Trusted Computing

Solving the Discrete Logarithm Problem Using P systems

Ma Xiaojing¹, Li Zhitang², Tu Hao²

¹ Computer Department of Huazhong University of Science and Technology, 430074, Wuhan, Peoples Republic of China
{lindahust, leeying, }@mail.hust.edu.cn

² Computer and Network Center, Huazhong University of Science and Technology, 430074, Wuhan, Peoples Republic of China

[M. Xiaojing, L. Zhitang, T. Hao, 2009]

The p system model used

- Active membranes:
rules of types a) - f)
- A priority relation on the rules:
if $R1 > R2$ and $R1$ is applicable then $R2$ is not used
- The result of the computation:
the number of a certain object sent out into the environment

The problem

Given a primitive root a of a prime p and an integer b , find x such that:

$$a^x \equiv b \pmod{p} \quad 1 \leq x, b \leq p-1$$

The P system

$$\Pi = (\tilde{V}, H, \tilde{\mu}, \omega_1, \omega_2, \omega_3, R, \rho, X)$$

$$V = \{A, B, C, D, F, G, H, H', X, \lambda\} \cup \{k_i, t_i, f_i \mid 1 \leq i \leq \lceil \log(p-1) \rceil\}$$

$$H = \{1, 2, 3\}$$

$$\mu = [{}^0_3[{}^0_2[{}^0_1]_1]_2]_3$$

$$\omega_1 = \{C, X, G^b\} \cup \{k_i \mid 1 \leq i \leq \lceil \log(p-1) \rceil\}$$

$$\omega_2 = \{A^a\}$$

$$\omega_3 = \emptyset$$

Initially: [[A..A [C X G..G k₁ k₂ ...]]]

The rules of the system

$$R_1 : A[_1]_1^0 \rightarrow [_1BD]_1^0$$

$$R_2 : [_1BC \rightarrow C]_1^0$$

$$R_3 : [_1D]_1^0 \rightarrow [_1D]_1^0 E$$

$$R_4 : [_1k_i]_1^0 \rightarrow [_1t_i]_1^+ [_1f_i]_1^- \text{ for } 1 \leq i \leq \lceil \log(p-1) \rceil$$

$$R_5 : [_1Ct_i \rightarrow C]_1^+ \text{ for } 1 \leq i \leq \lceil \log(p-1) \rceil$$

$$R_6 : [_1X \rightarrow XX]_1^+$$

$$R_7 : [_1Cf_i \rightarrow CX]_1^- \text{ for } 1 \leq i \leq \lceil \log(p-1) \rceil$$

$$R_8 : [_1X \rightarrow XX]_1^-$$

$$R_{17} : [_1FH \rightarrow FGH']_1^0$$

$$R_{18} : [_1H]_1^0 \rightarrow \lambda$$

$$R_{19} : [_1H']_1^0 \rightarrow [_1]_1^0 A$$

$$R_9 : [_1D \rightarrow \lambda]_1^+$$

$$R_{10} : [_1D \rightarrow \lambda]_1^-$$

$$R_{11} : E[_1]_1^0 \rightarrow [_1F]_1^0$$

$$R_{12} : [_1F \rightarrow F^a]_1^-$$

$$R_{13} : [_2[_1]_1^+ [_1]_1^-]_2^0 \rightarrow [_2[_1]_1^0]_2^0 [_2[_1]_1^0]_2^0$$

$$R_{14} : [_1F^p \rightarrow \lambda]_1^0$$

$$R_{15} : [_1FG \rightarrow H]_1^0$$

$$R_{16} : [_1GH \rightarrow G^2H']_1^0$$

$$R_{20} : [_1F]_1^0 \rightarrow [_1]_1^0 A$$

$$R_{21} : [_2X]_2^0 \rightarrow [_2]_2^0 X$$

$$R_{22} : [_3X]_3^0 \rightarrow [_3]_3^0 X$$

$$\rho = \{R_{1-3,14-19} > R_4, R_{16,17} > R_{18}, R_{14} > R_{15} > R_{20}\}$$

The computation 1.

Initially: $[_3 \ [_2 \ A\dots A \ [_1 \ C \ X \ G\dots G \ k_1 \ k_2 \ \dots] ^0] ^0] ^0$

Using rules:

$$R_1 : A[_1]_1^0 \rightarrow [_1 BD]_1^0$$

$$R_2 : [_1 BC \rightarrow C]_1^0$$

$$R_3 : [_1 D]_1^0 \rightarrow [_1 D]_1^0 E$$

we get: $[_3 \ [_2 \ [_1 BD \ BD \ \dots \ BD \ C \ X \ G\dots G \ k_1 \ k_2 \ \dots] ^0] ^0] ^0$

Then in *a* steps:

$$[_3 \ [_2 \ E\dots E \ [_1 \ D \ \dots \ D \ C \ X \ G\dots G \ k_1 \ k_2 \ \dots] ^0] ^0] ^0$$

The computation 2.

$$[{}_3 [{}_2 E \dots E [{}_1 D \dots D C X G \dots G k_1 k_2 \dots]^0]^0]^0$$

Using rules:

$$R_4 : [{}_1 k_i]_1^0 \rightarrow [{}_1 t_i]_1^+ [{}_1 f_i]_1^- \text{ for } 1 \leq i \leq \lceil \log(p-1) \rceil$$

$$R_{11} : E[{}_1]_1^0 \rightarrow [{}_1 F]_1^0$$

we get:

$$[{}_3 [{}_2 [{}_1 F \dots F D \dots D C X G \dots G t_1 k_2 \dots]^+ [{}_1 F \dots F D \dots D C X G \dots G f_1 k_2 \dots]^-]^0]^0$$

The computation 3.

$$\left[\begin{array}{c} [3 \quad [2 \quad [1 \quad F \dots F \quad D \quad \dots \quad D \quad C \quad X \quad G \dots G \quad t_1 \quad k_2 \quad \dots] ^+ \\ [1 \quad F \dots F \quad D \quad \dots \quad D \quad C \quad X \quad G \dots G \quad f_1 \quad k_2 \quad \dots] ^- \end{array} \right]^0]^0$$

Using rules:

- $R_5 : [1 C t_i \rightarrow C]_1^+$ for $1 \leq i \leq \lceil \log(p-1) \rceil$
- $R_6 : [1 X \rightarrow XX]_1^+$
- $R_7 : [1 C f_i \rightarrow CX]_1^-$ for $1 \leq i \leq \lceil \log(p-1) \rceil$
- $R_8 : [1 X \rightarrow XX]_1^-$
- $R_9 : [1 D \rightarrow \lambda]_1^+$
- $R_{10} : [1 D \rightarrow \lambda]_1^-$
- $R_{12} : [1 F \rightarrow F^a]_1^-$
- $R_{13} : [2 [1]_1^+ [1]_1^-]_2^0 \rightarrow [2 [1]_1^0]_2^0 [2 [1]_1^0]_2^0$

we get: $\left[\begin{array}{c} [3 \quad [2 \quad [1 \quad F \dots F \quad C \quad XX \quad G \dots G \quad k_2 \quad k_3 \quad \dots] ^0 \\ [2 \quad [1 \quad F \dots F \quad C \quad XXX \quad G \dots G \quad k_2 \quad k_3 \quad \dots] ^0 \end{array} \right]^0]^0$

The computation 4.

$$\begin{aligned} & [{}_3 [{}_2 [{}_1 F \dots F \ C \ XX \ G \dots G \ k_2 \ k_3 \ \dots]^0]^0]^0 \\ & [{}_2 [{}_1 F \dots F \ C \ XXX \ G \dots G \ k_2 \ k_3 \ \dots]^0]^0]^0 \end{aligned}$$

Using the rule:

$$R_{14} : [{}_1 F^P \rightarrow \lambda]_1^0$$

we get:

$$\begin{aligned} & [{}_3 [{}_2 [{}_1 \mathbf{F} \dots \mathbf{F} \ C \ XX \ G \dots G \ k_2 \ k_3 \ \dots]^0]^0]^0 \\ & [{}_2 [{}_1 \mathbf{F} \dots \mathbf{F} \ C \ XXX \ G \dots G \ k_2 \ k_3 \ \dots]^0]^0]^0 \end{aligned}$$

The computation 5.

$$\begin{aligned}
 & [{}_{3} [{}_{2} [{}_{1} F\dots F \ C \ XX \ G\dots G \ k_2 \ k_3 \ \dots]{}^0 \]{}^0 \\
 & \quad [{}_{2} [{}_{1} F\dots\dots F \ C \ XXX \ G\dots G \ k_2 \ k_3 \ \dots]{}^0 \]{}^0 \]{}^0
 \end{aligned}$$

Using:

$$R_{15} : [{}_{1} FG \rightarrow H]{}^0_1$$

F s and G s disappear, if $a^2=b \pmod p$ or $a^3=b \pmod p$. In this case $R_{18} : [{}_{1} H]{}^0_1 \rightarrow \lambda$ dissolves membrane 1,

and $R_{21} : [{}_{2} X]{}^0_2 \rightarrow [{}_{2}]{}^0_2 X$

$R_{22} : [{}_{3} X]{}^0_3 \rightarrow [{}_{3}]{}^0_3 X$ outputs the X s.

The computation 6.

Otherwise the F s and G s are changed back by

$$R_{16} : [{}_1GH \rightarrow G^2H']_1^0 \quad R_{19} : [{}_1H']_1^0 \rightarrow [{}_1]_1^0 A$$

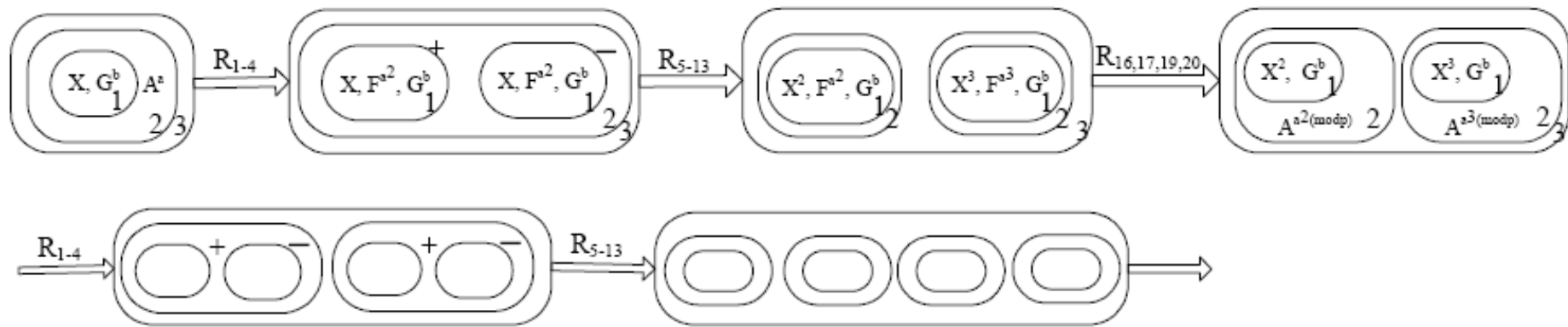
$$R_{17} : [{}_1FH \rightarrow FGH']_1^0 \quad R_{20} : [{}_1F]_1^0 \rightarrow [{}_1]_1^0 A$$

and we get

$$\begin{aligned} & [{}_3 [{}_2 A \dots A [{}_1 C \ XX \ G \dots G \ k_2 \ k_3 \ \dots]_1^0 \]_2^0 \\ & [{}_2 A \dots A [{}_1 C \ XXX \ G \dots G \ k_2 \ k_3 \ \dots]_1^0 \]_2^0 \]_3^0 \end{aligned}$$

and the process can be repeated.

The process of the computation



We get the solution to $a^x \equiv b \pmod{p}$ $1 \leq x, b \leq p-1$ in at most $\log p$ loops.

References

1. Gh. Paun. P systems with active membranes attacking NP-complete problems. *Journal of Automata Languages and Combinatorics* 6 2001 75-90.
2. M.A. Gutierrez-Naranjo, M.J. Perez-Jimenez, F.J. Romero-Campero. A uniform solution to SAT using membrane creation. *Theoretical Computer Science* 371 2007 54-61.
3. M. Xiaojing, L. Zhitang, T. Hao. Solving the discrete logarithm problem using P systems. *Proc. NSWCTC '09, the 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing - Volume 02*. IEEE Computer Society Washington DC, USA, 438-441.
4. Gh. Paun, G. Rozenberg, A. Salomaa, editors. *The Oxford Handbook of Membrane Computing*. Oxford University Press, 2007.