Membrane Systems:
A Molecular Model for Computing

Kamala Krithivasan

Theoretical Computer Science Laboratory,

Department of Computer Science and Engineering,

Indian Institute of Technology Madras,

Chennai – 600036.

Email: kamala@iitm.ac.in

Organization

- Models of Computing
- Membrane Systems
- Motivation
- Our Contribution
- Future Direction

Models of Computing

Conventional

Unconventional

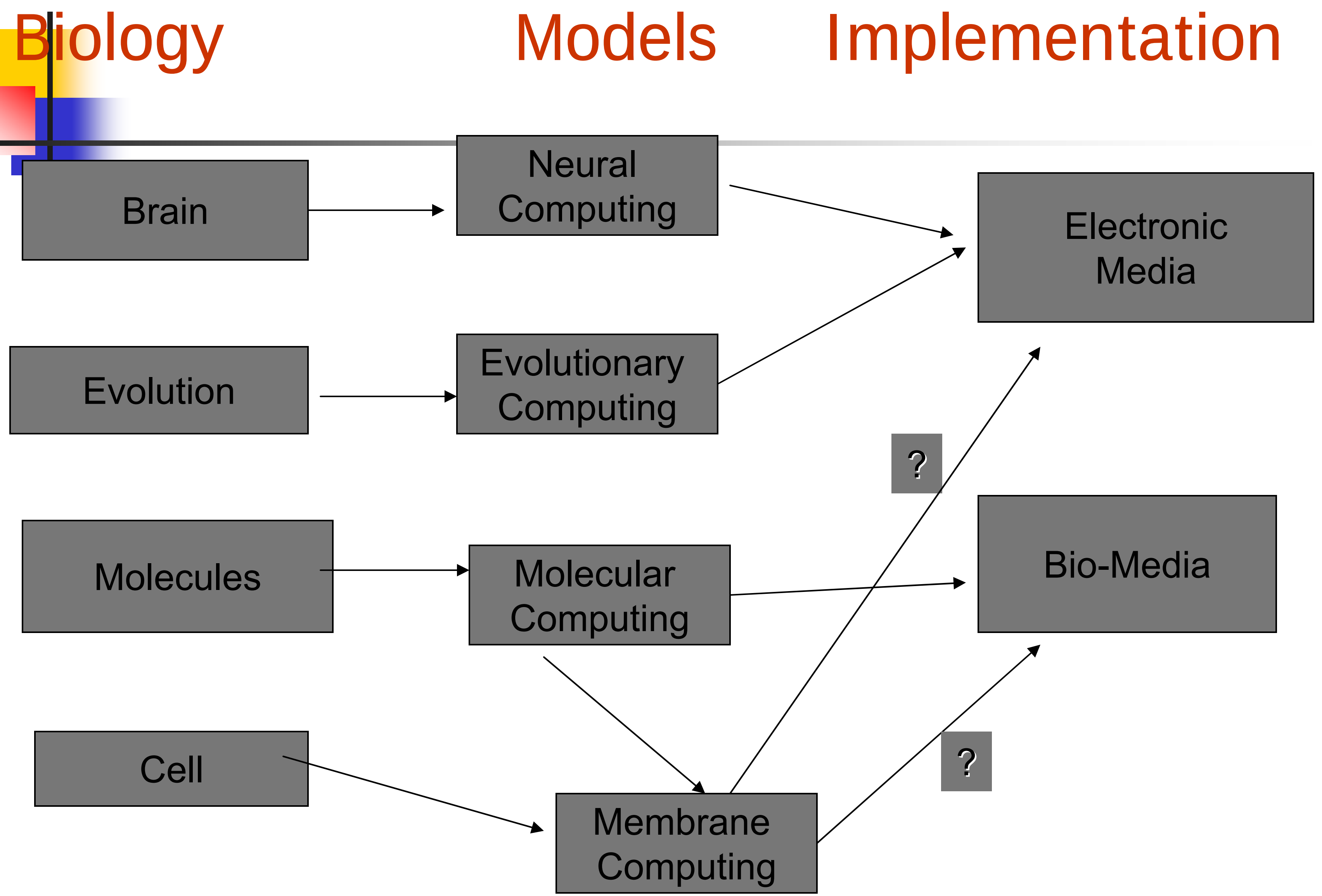
Why we need unconventional models?

- ❖ NP-Complete Problems – No efficient algorithms in classical computers.
- ❖ Classical computers are deterministic.
- ❖ Parallelism is restricted in classical computers.
- ❖ Exhaustive search is exponential in classical computers.

"The processes which take place in a cell, the reactions which develop in cell regions, the processing of substances, energy and information in these regions and through the membranes which delimit them, are computational processes."

D. Bray
Protein molecules as computational elements in living cells.
Nature, 376 (1995), 307-312.

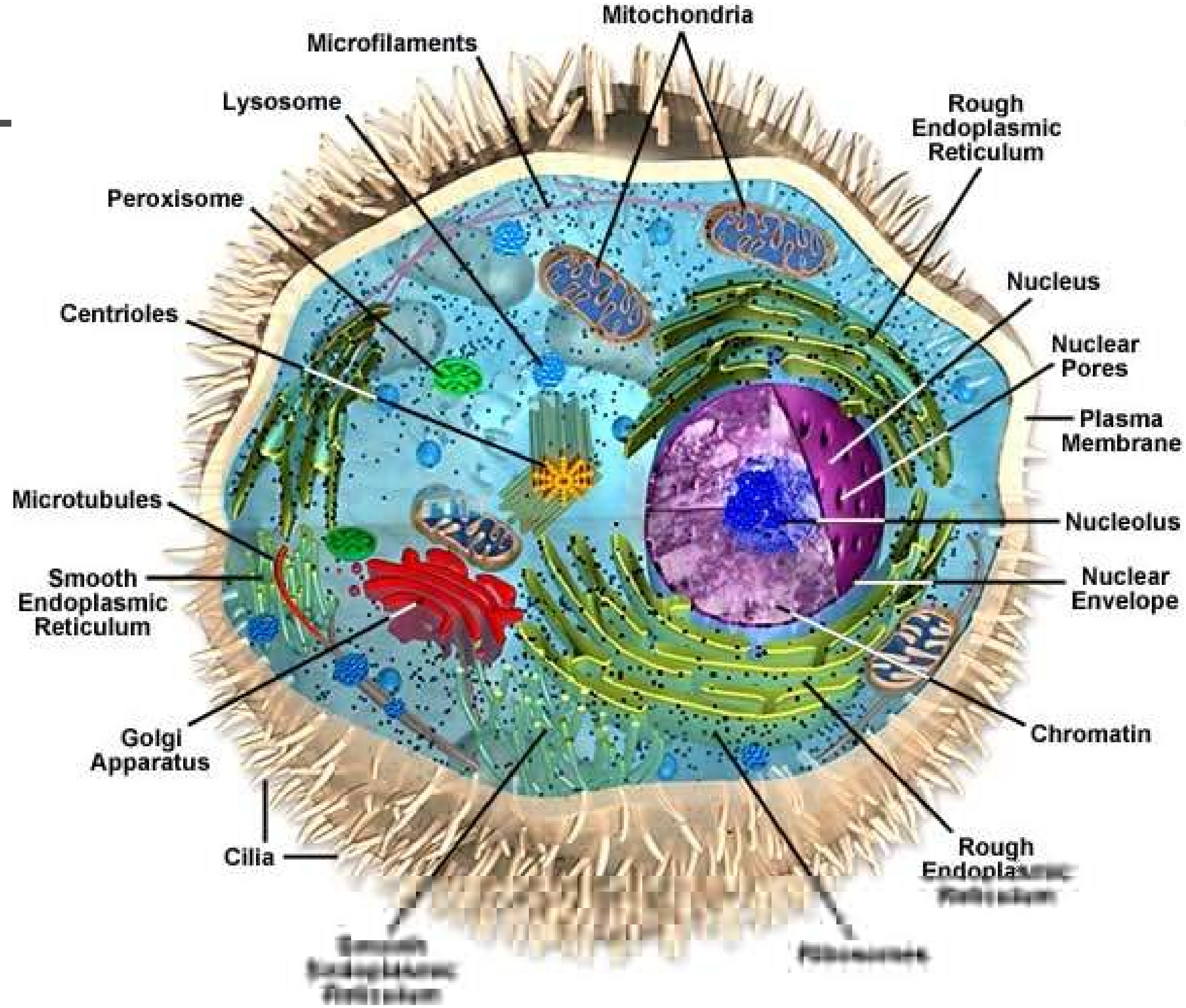
Bio-Domains of Natural Computing



Cell Biology

- A cell has a complex structure, with several compartments delimited inside the main membrane by several inner membranes: the nucleus, the Golgi apparatus, several vesicles, etc.
- In principle, all these membranes are similar, so we consider the **plasma membrane** and consider its structure and functions.

Animal Cell



Mitochondria
Microfilaments
Lysosome
Rough Endoplasmic Reticulum
Peroxisome
Nucleus
Centrioles
Nuclear Pores
Plasma Membrane
Microtubules
Nucleolus
Smooth Endoplasmic Reticulum
Nuclear Envelope
Golgi Apparatus
Chromatin
Cilia
Rough

Membrane Systems

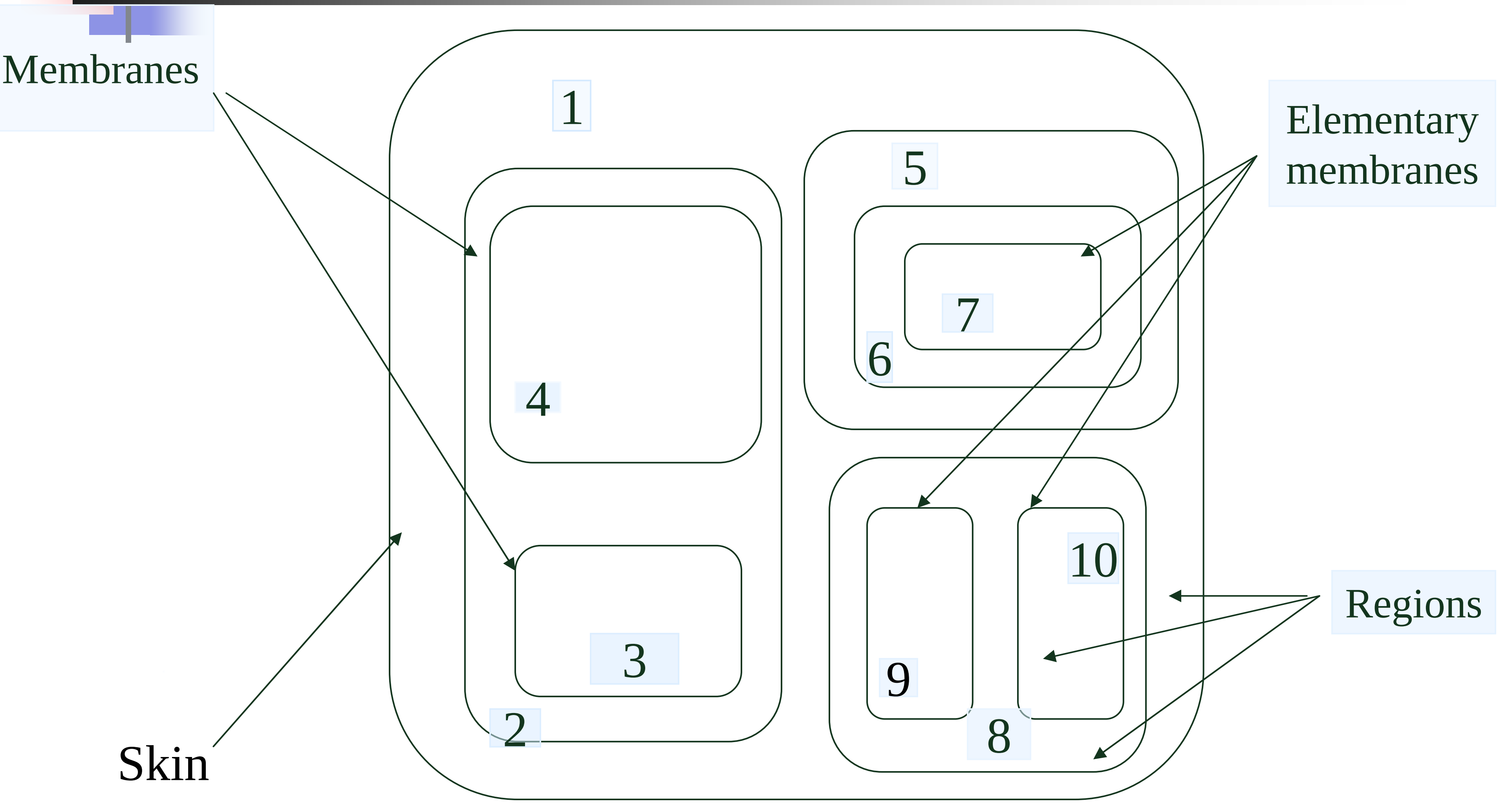
- New field of research, motivated by the way *nature* computes at the *cellular* level, introduced by Prof. Gh. Păun. It is also called as **P systems.**

- A class of distributed parallel computing devices of biochemical type.

- The three fundamental features of cells which will be used in our computing model are:

    - The **membrane structure**, (where)
    - **multisets** of chemical compounds (evolve according to)
    - (prescribed) **rules.**

Membrane Structure



Objects

- Objects can be considered as *atomic* or they can have *structure*.
- Symbol-objects
- String-objects
- Array-objects
- Graph-objects

Evolution Rules

❖ Generally, we consider **context-free** [1] rules for processing both symbol-objects and string-objects.

❖ For string-objects, we may consider **splicing** [2] rules also.

❖ P systems with symbol-objects.

❖ Rewriting P systems.

❖ Splicing P systems.

[1] J.E. Hopcroft, J.D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, 1979.

[2] Gh. Păun, G. Rozenberg, A. Salomaa, *DNA Computing. New Computing Paradigms*. Springer-Verlag, Berlin, 1998.

Formal Definition

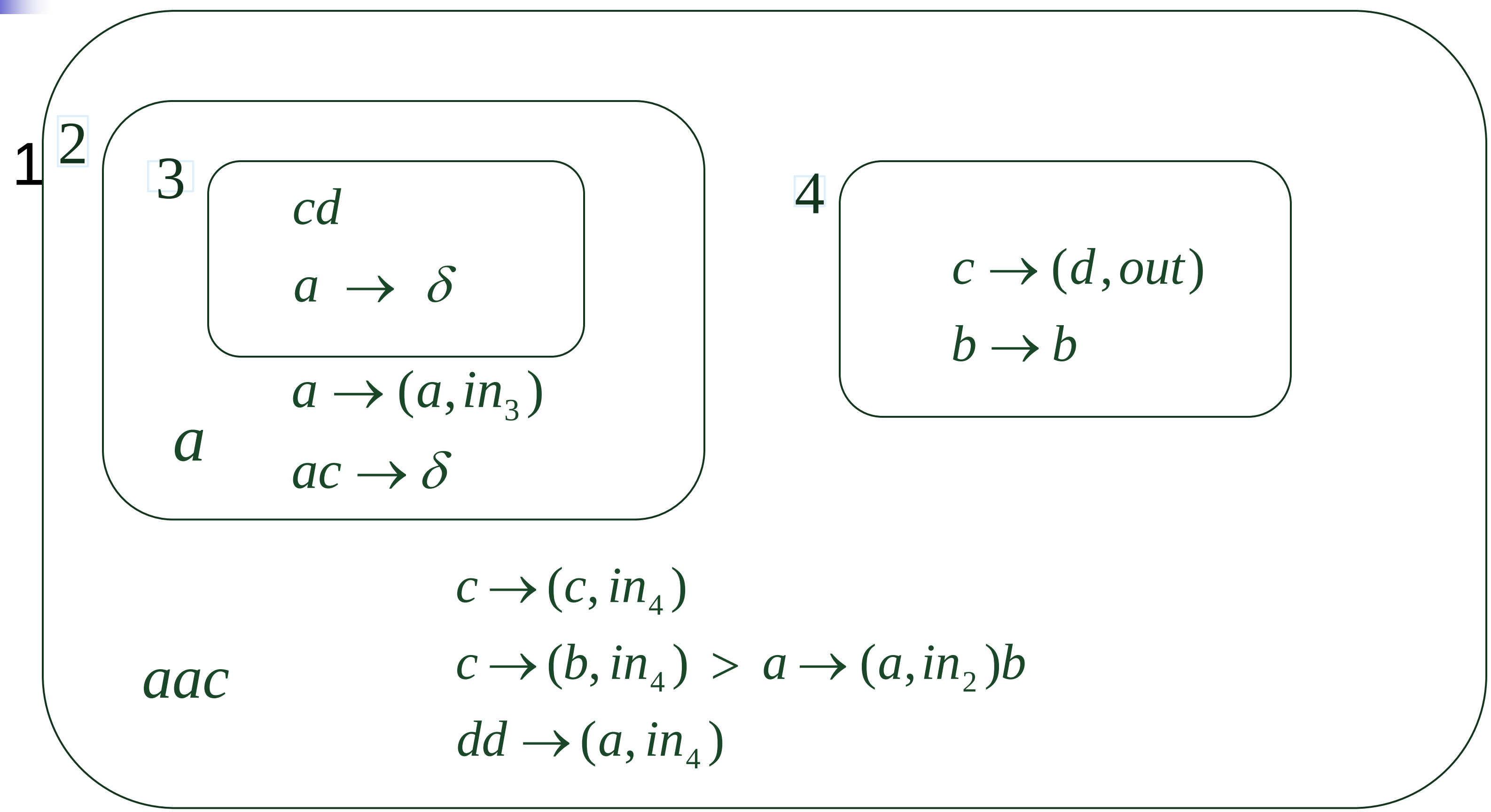
P system of degree *n, n ≥ 1,* is a construct

$$\Pi = (V,T,C,\mu,w_1,\ldots,w_n,(R_1,\rho_1),\ldots(R_n,\rho_n),i_0),$$

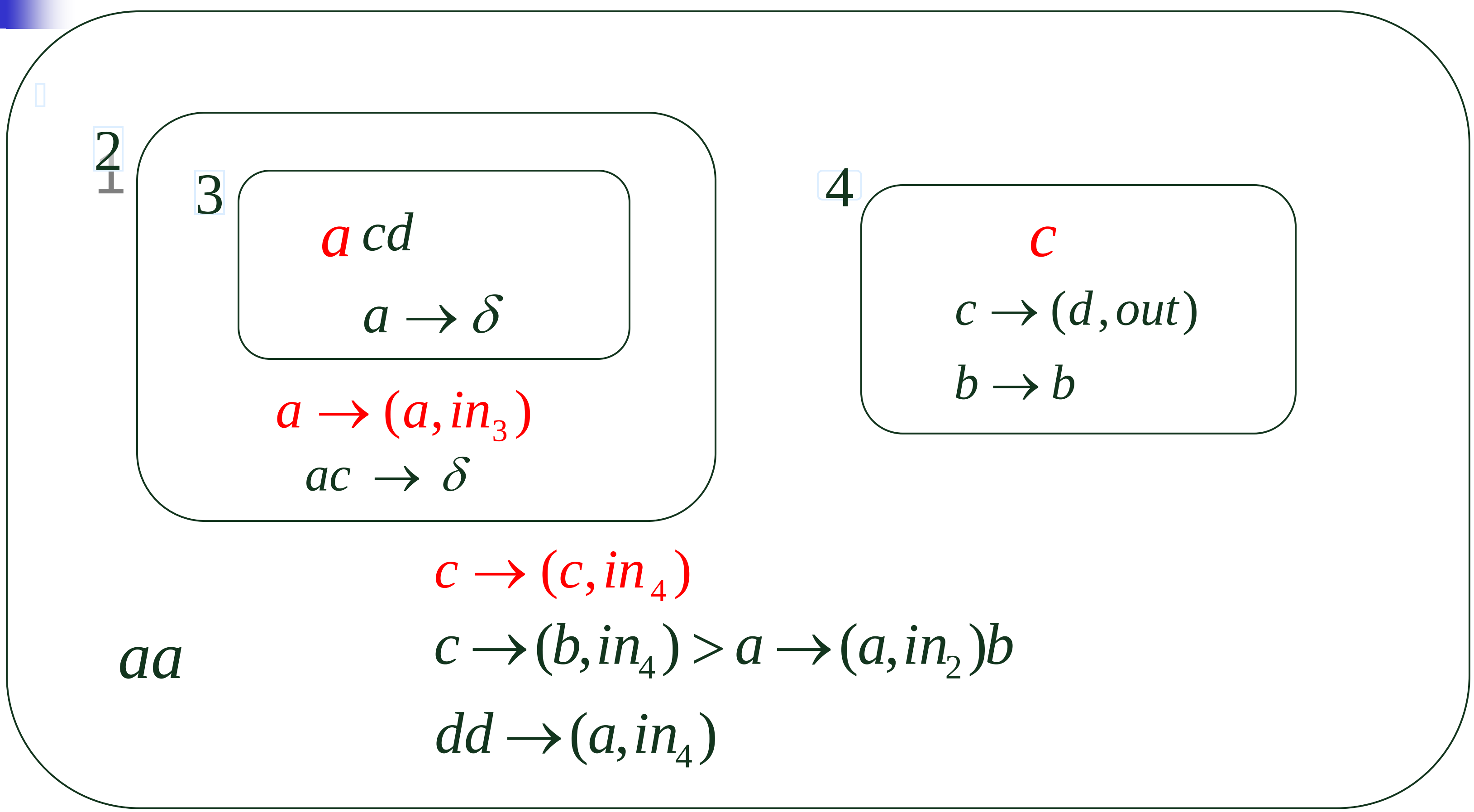
where:

- **V** is an alphabet; its elements are called *objects*;
- **T** $\subseteq$ **V** is an output alphabet;
- **C** $\subseteq$ **V**, **C** $\cap$ **T** = **Ø**, is a set of catalysts;
- μ is a membrane structure;
- $\mathbf{w_i}$, $1 \leq i \leq n$, is a multiset of objects over **V** present in region **i**;
- $\mathbf{R_i}$, $1 \leq i \leq n$, is a finite set of rules associated with region **i**;
- ρ_i, $1 \leq i \leq n$, is a partial order relation over $\mathbf{R_i}$;
- $\mathbf{i_0}$ is an output membrane.

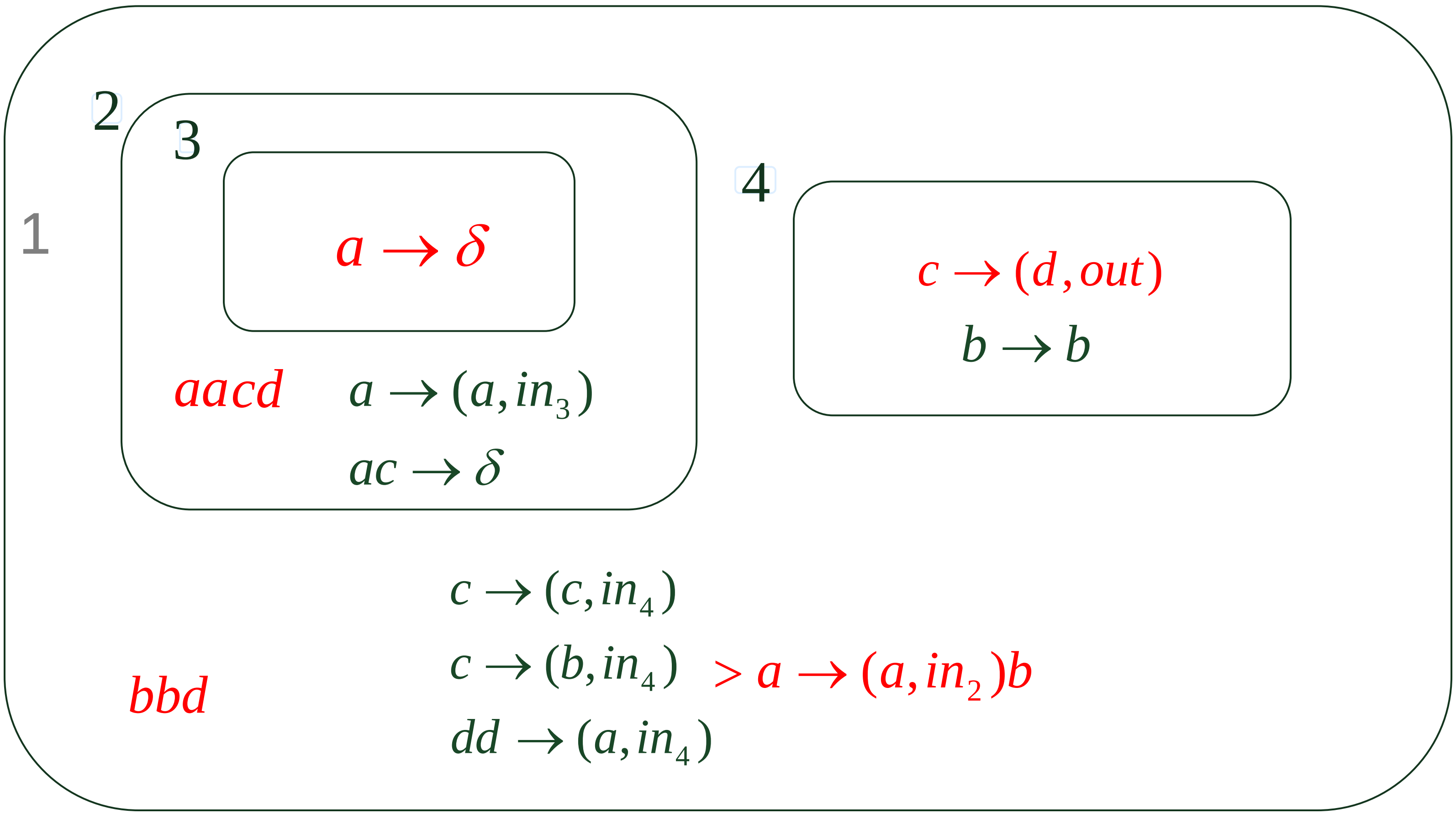
P Systems with Multisets of Objects



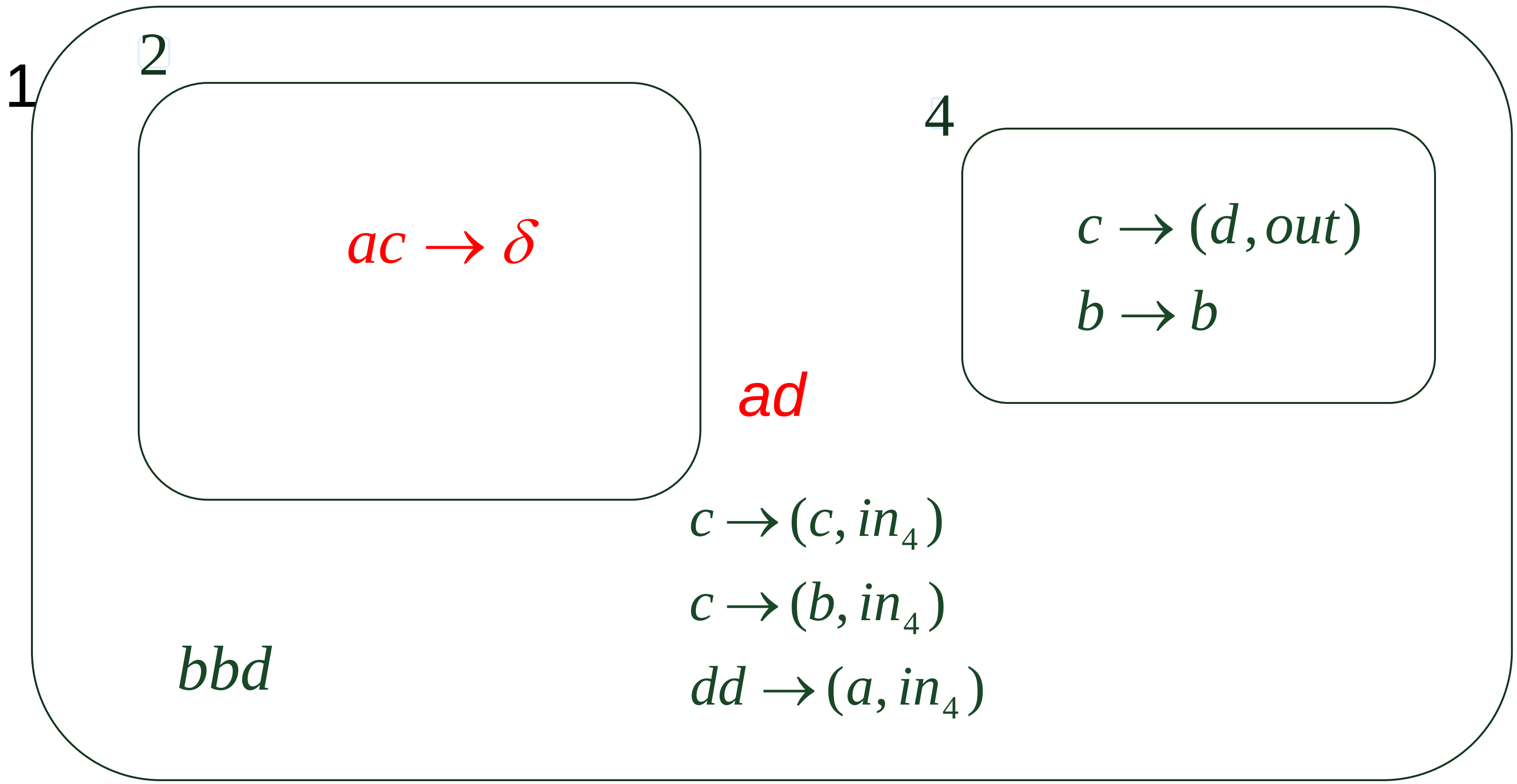
P Systems with Multisets of Objects



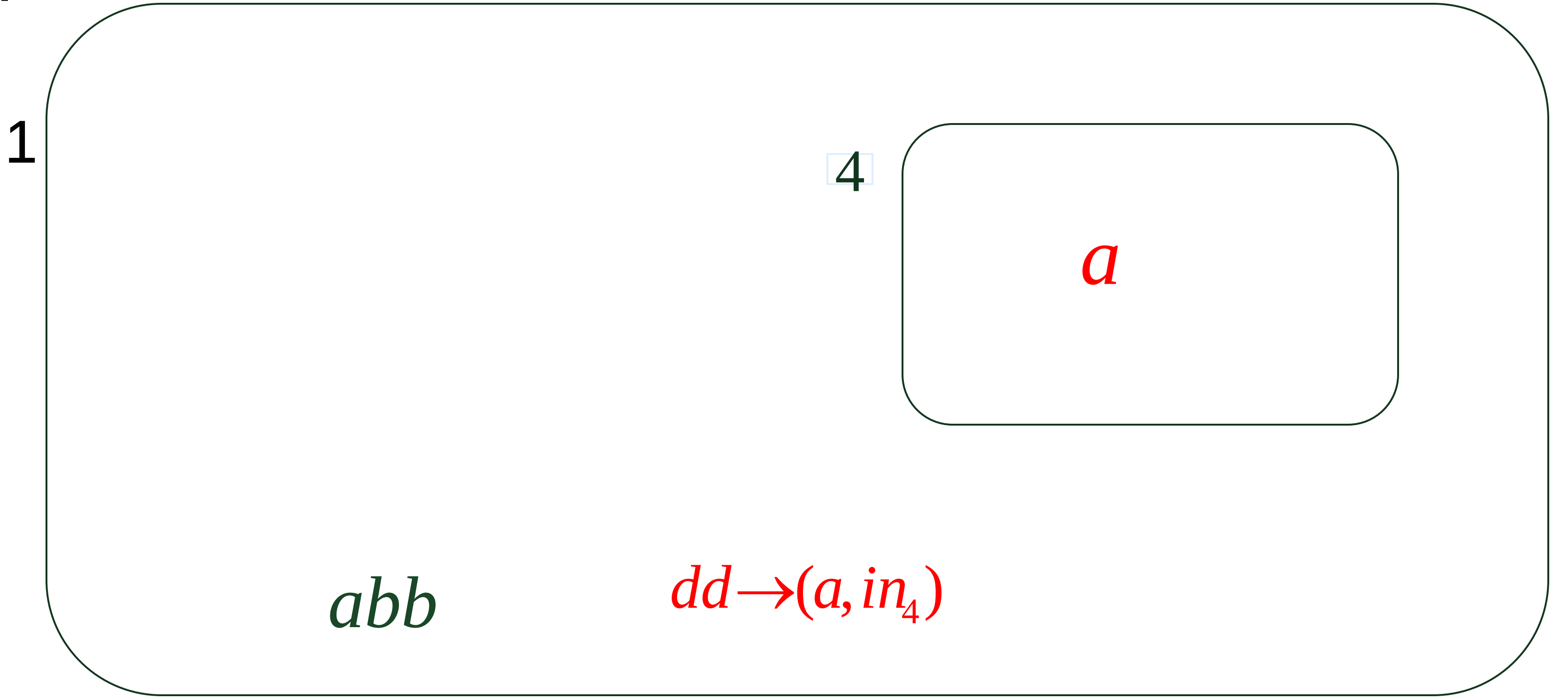
P Systems with Multisets of Objects



P Systems with Multisets of Objects



P Systems with Multisets of Objects

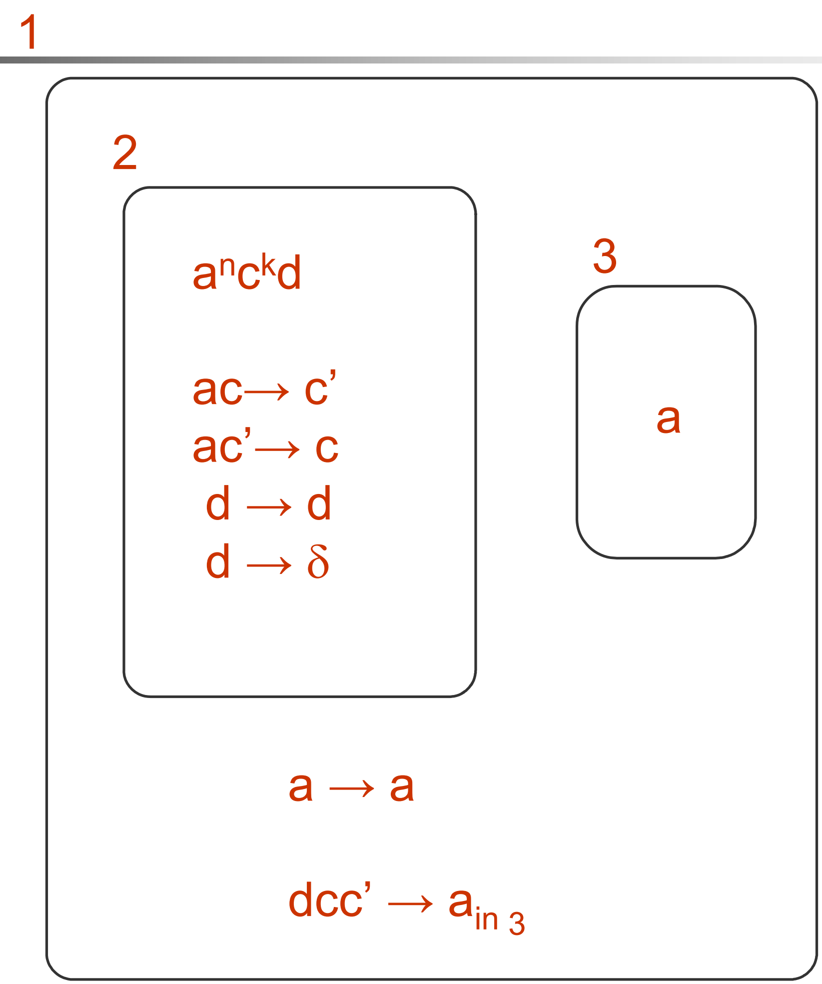


Formal Definition (Contd...)

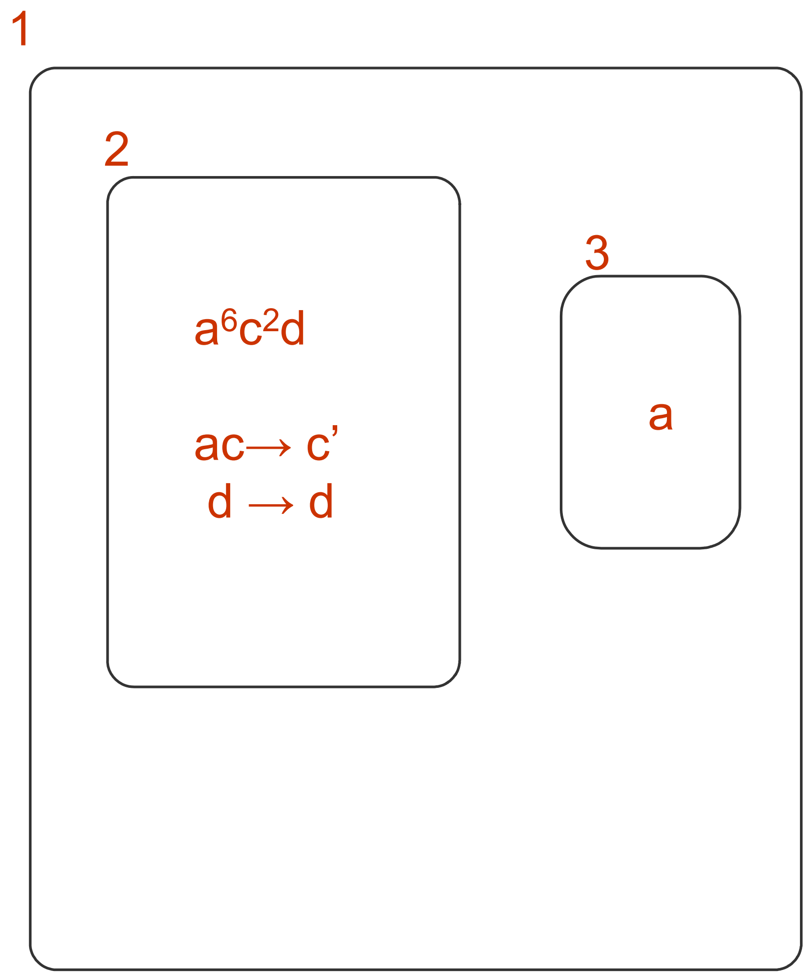
- The rules are of the following form:
  - For symbol-objects: $u \rightarrow v$, where u is a string over **V** and $v = v'$ or $v = v'\delta$, where v' is a string over
    $$\{a_{here}, a_{out}, a_{in} \mid a \in \mathbf{V}\},$$
    and δ is a special symbol not in **V**;
  - For rewriting P systems: $X \rightarrow v(tar)$, where $X \rightarrow v$ is a context-free rule and $tar \in \{here, out, in\}$;
  - For splicing P systems: $(r; tar_1, tar_2)$, where $r = u_1\#u_2\$u_3\#u_4$ is a splicing rule over **V** and $tar_1, tar_2 \in \{here, out, in\}$;

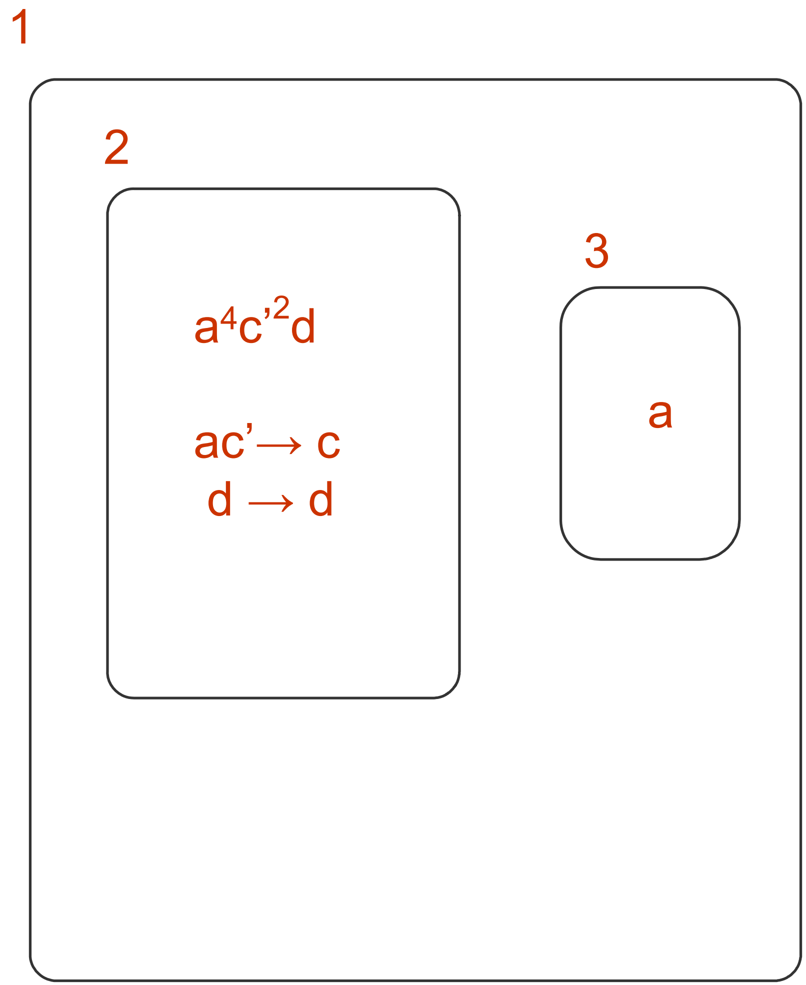
P systems can be viewed as

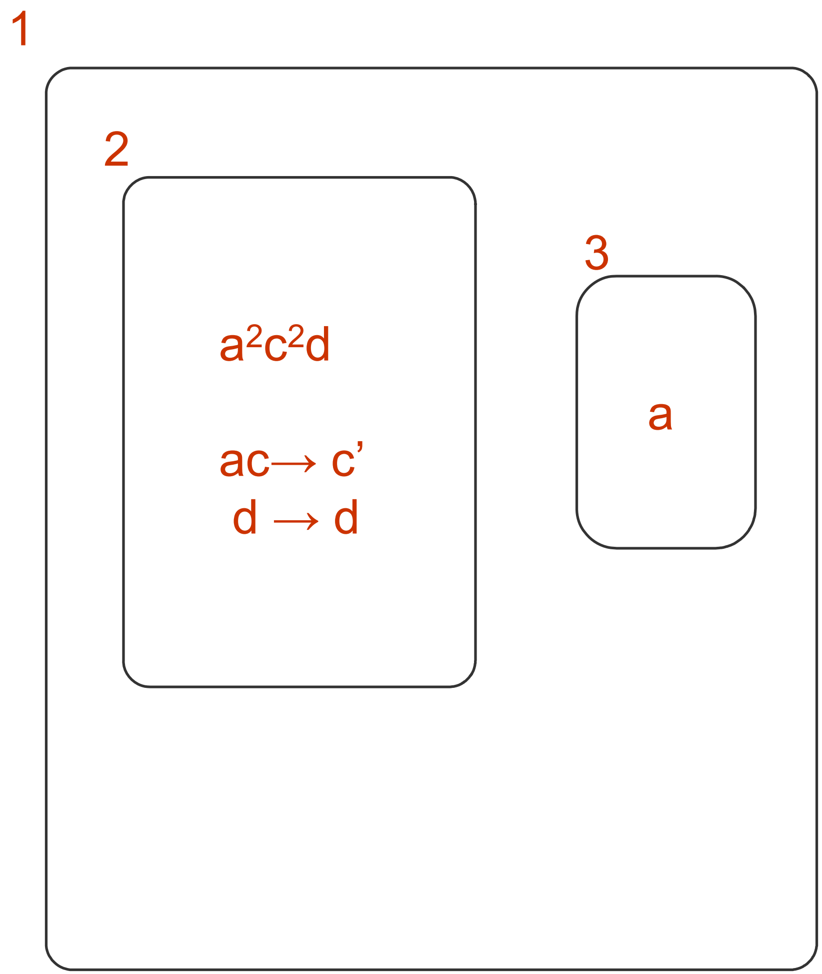
- Generative devices
- Computing devices
- Decidability devices

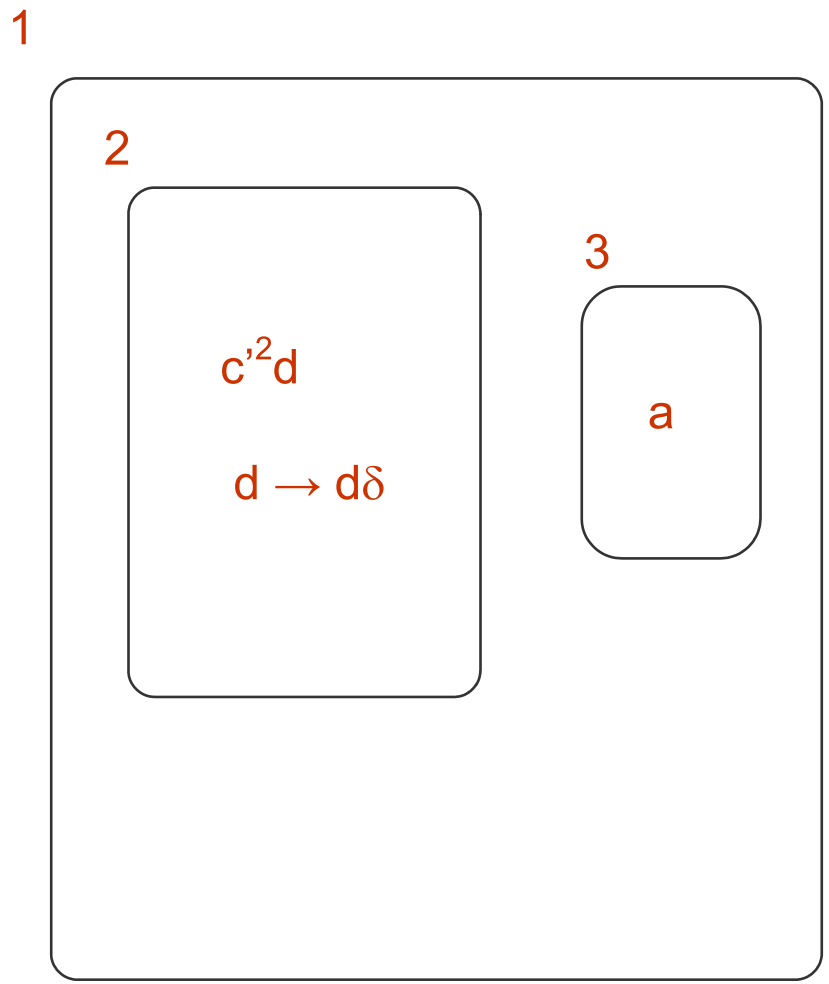


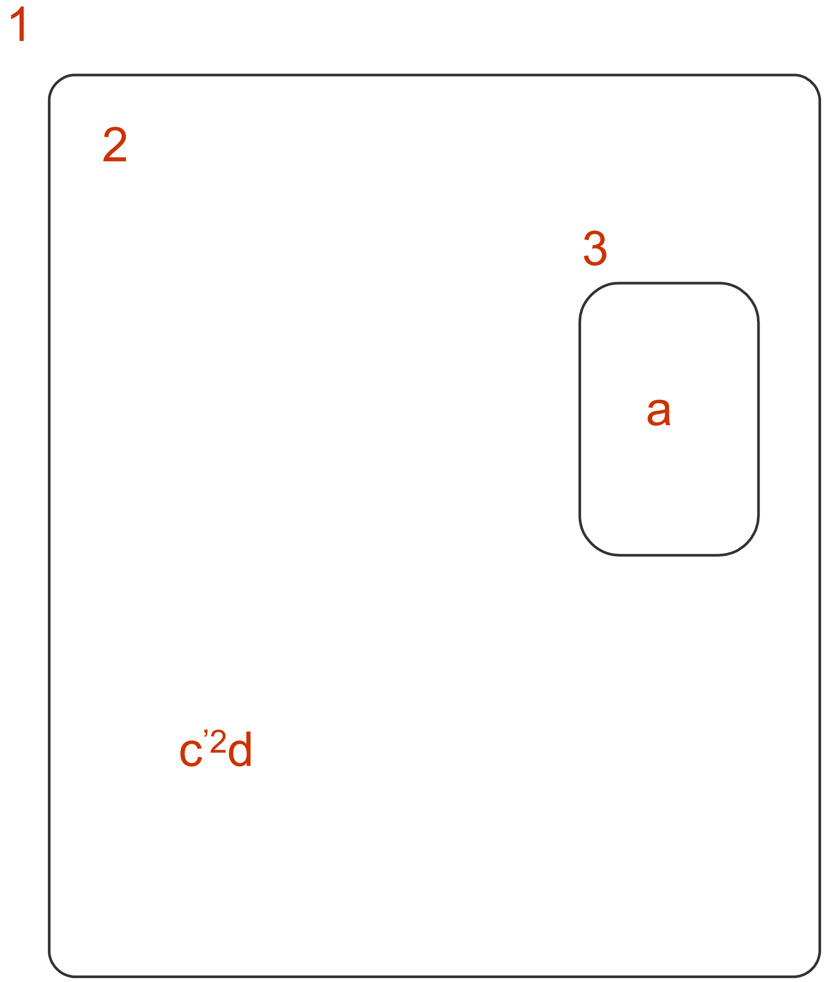
A P system deciding whether k divides n

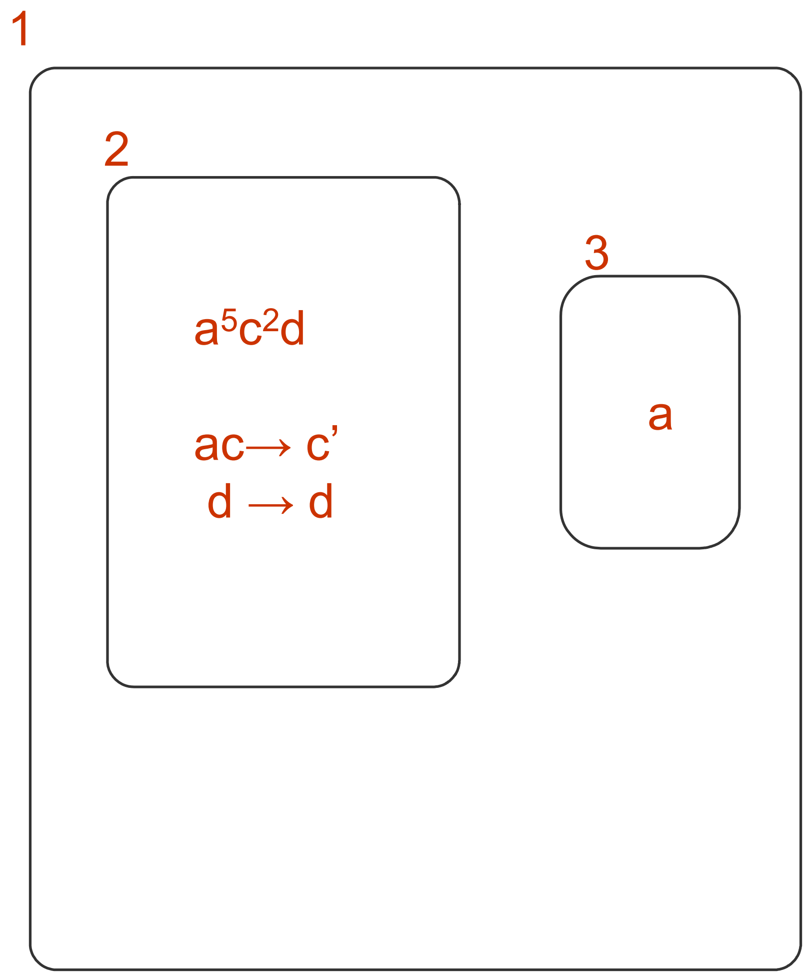

1
2
a⁶c²d
ac→ c’
d → d
3
a


1
2
$a^4c'^2d$
$ac' \to c$
$d \to d$
3
a

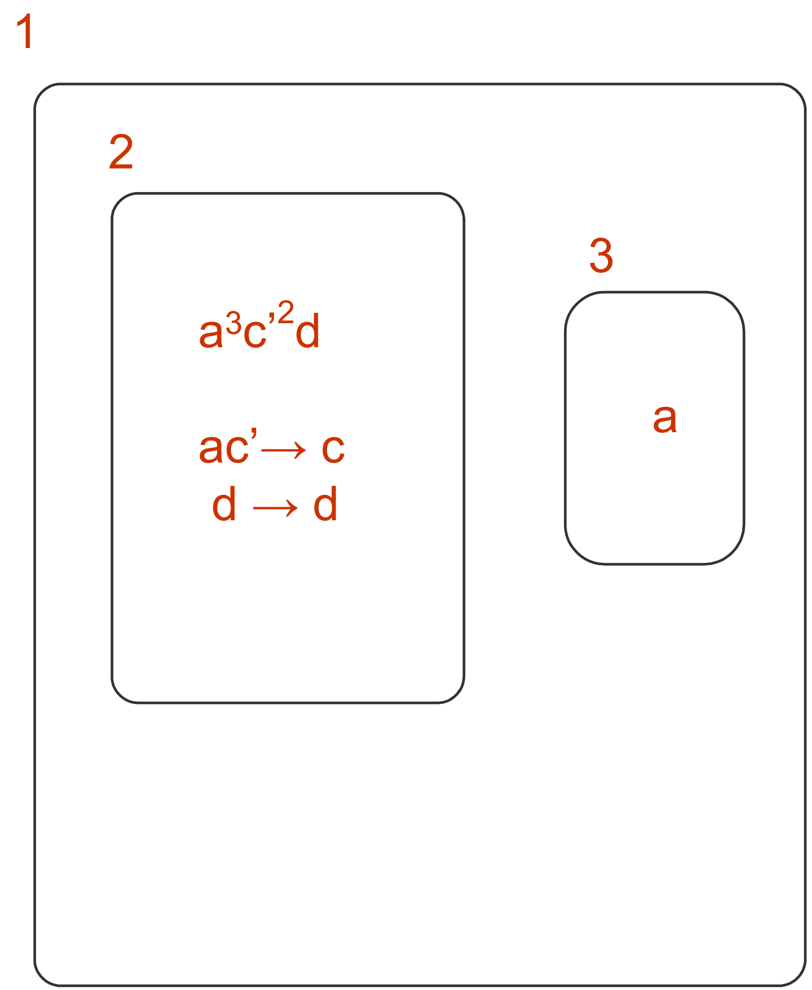

1
2
a²c²d
ac→ c'
d → d
3
a

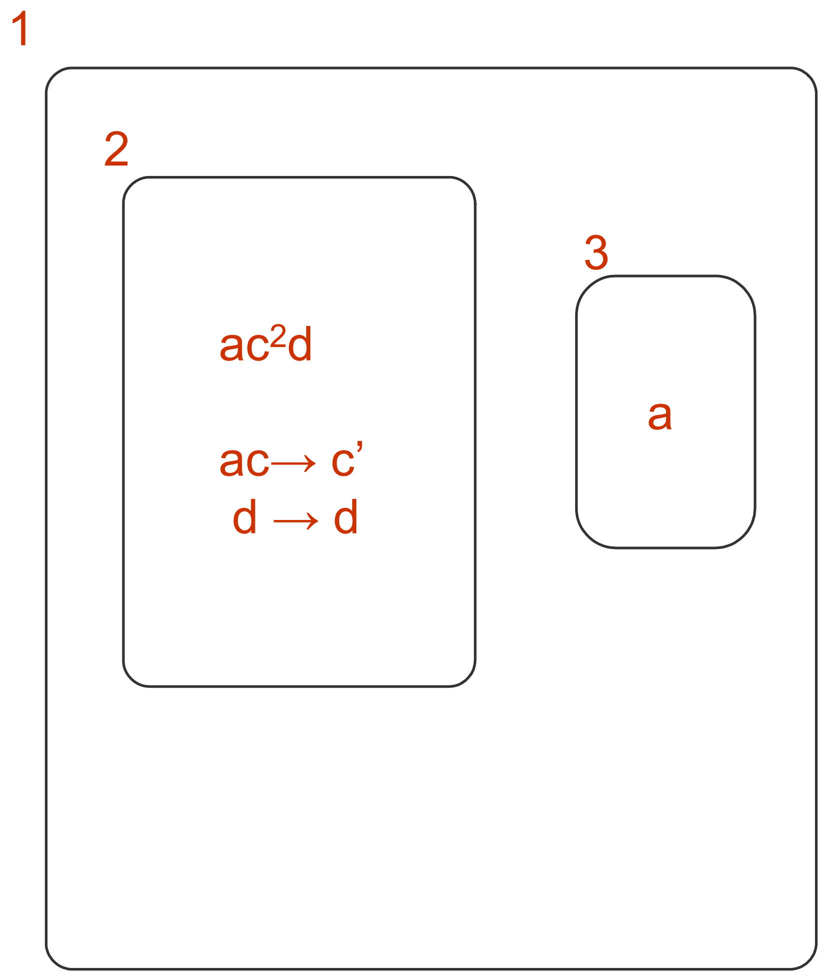

1
2
c’2d
d → dδ
3
a


1
2
3
a
c'2d

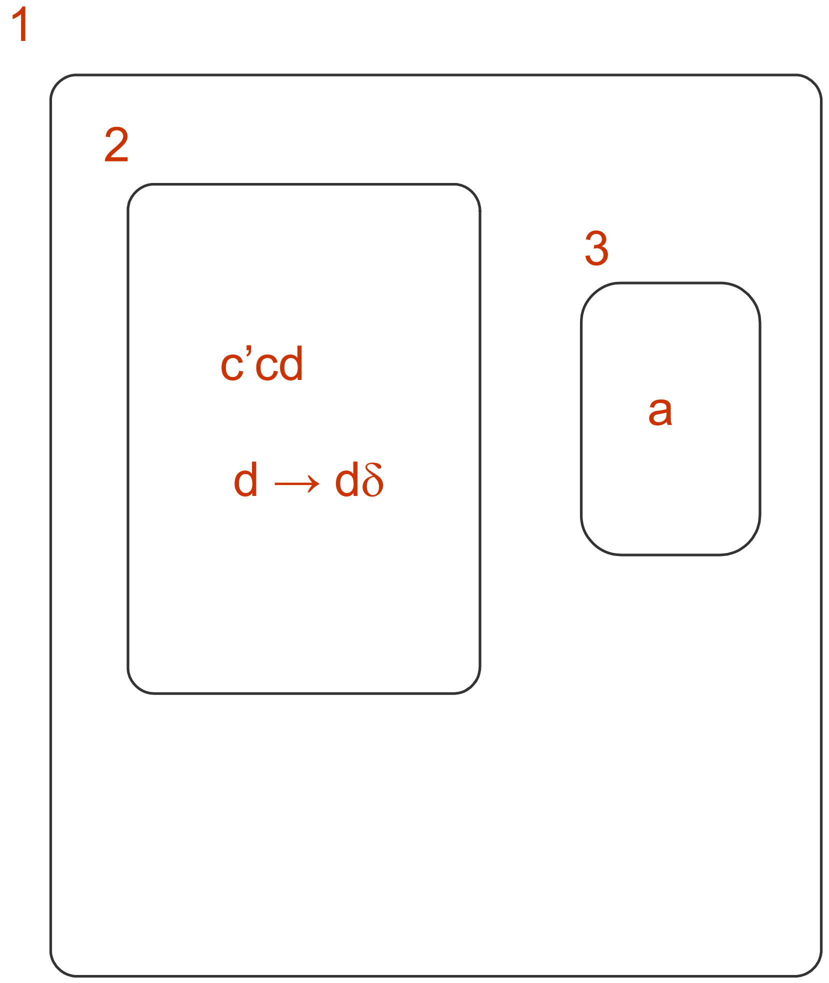


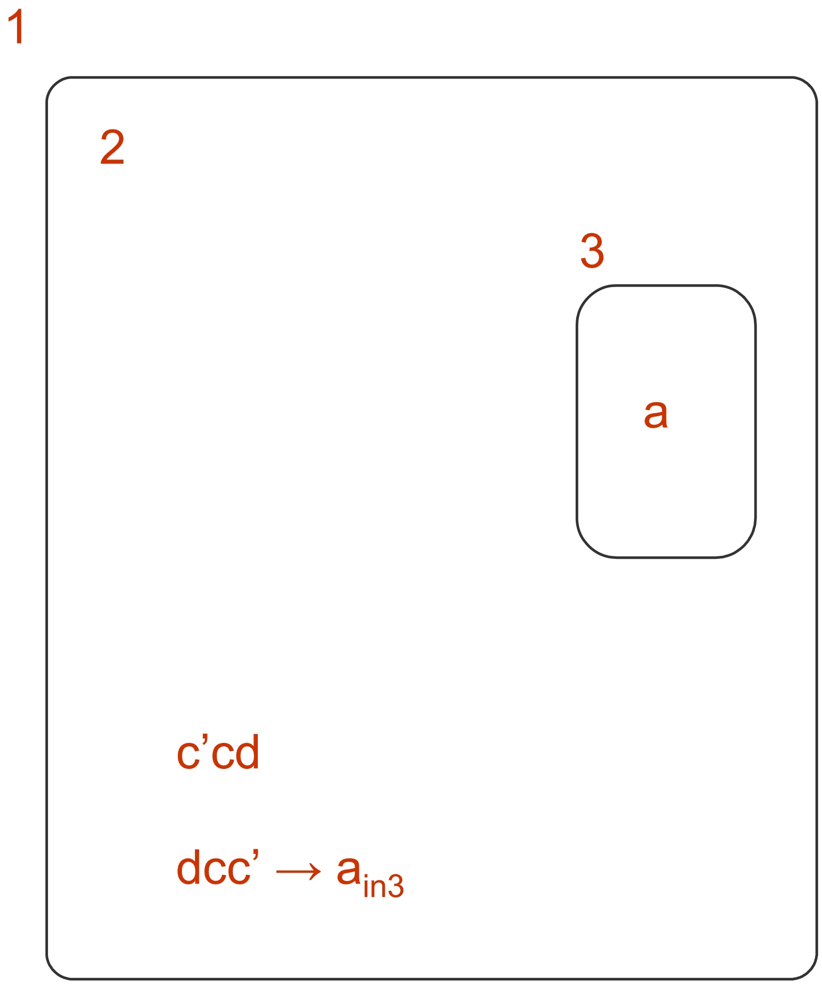
1
2
a5c2d
ac→ c'
d → d
3
a

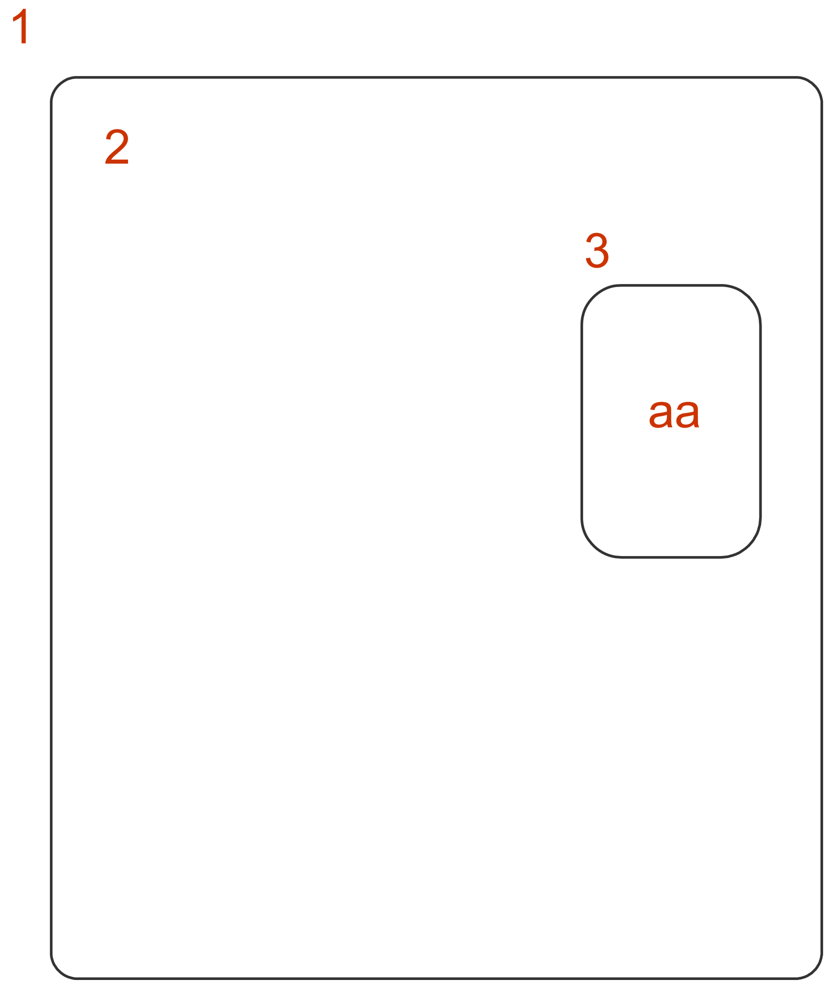

1
2
$a^3c'^2d$
$ac' \rightarrow c$
$d \rightarrow d$
3
a



1
2
ac²d
ac→ c'
d → d
3
a

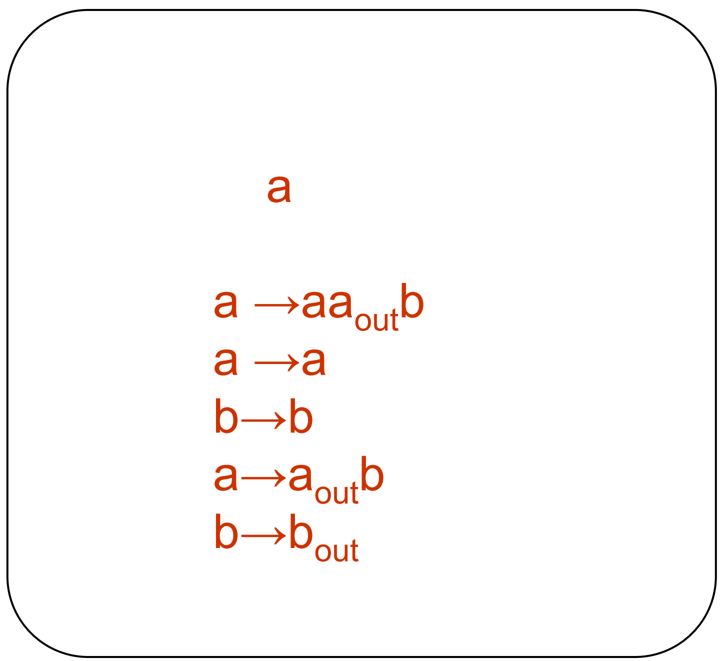

1
2
c'cd
d → dδ
3
a


1
2
3
a
c'cd
dcc' → a_{in3}


1
2
3
aa

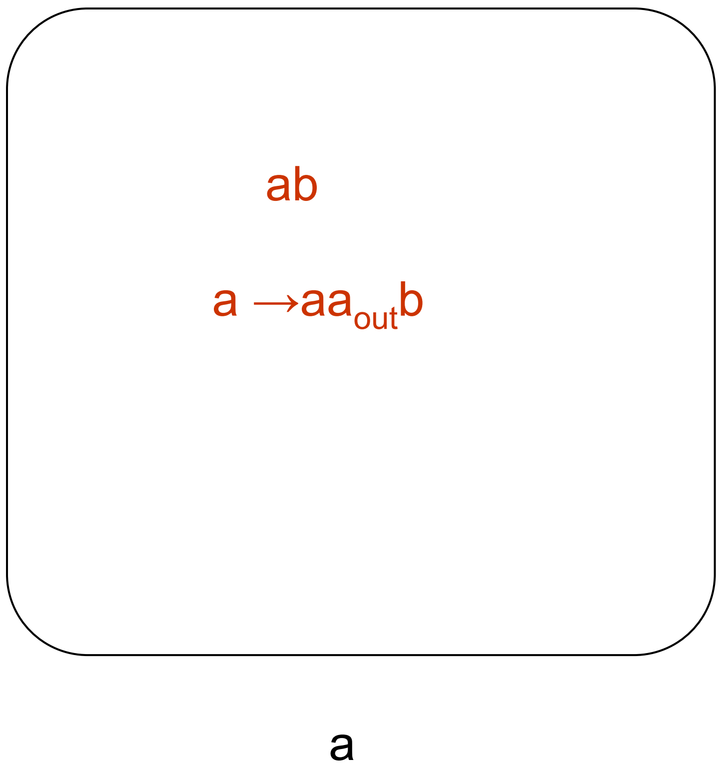
P system generating the Dyck set

$\pi_4 = (\{a,b\}, \{a,b\}, \phi, [_1]_1, a, (R_1, \phi), \infty)$,
$R_1 = \{ a \rightarrow aa_{out}b, a \rightarrow a, b \rightarrow b, a \rightarrow a_{out}b, b \rightarrow b_{out}\}$,



a

$a \rightarrow a a_{out} b$


ab
$a \rightarrow a a_{out} b$
a

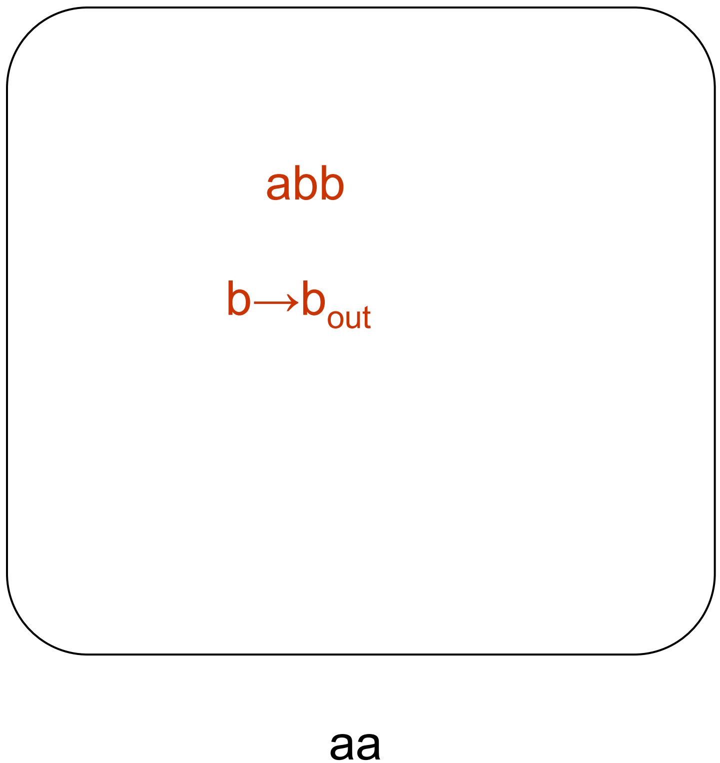
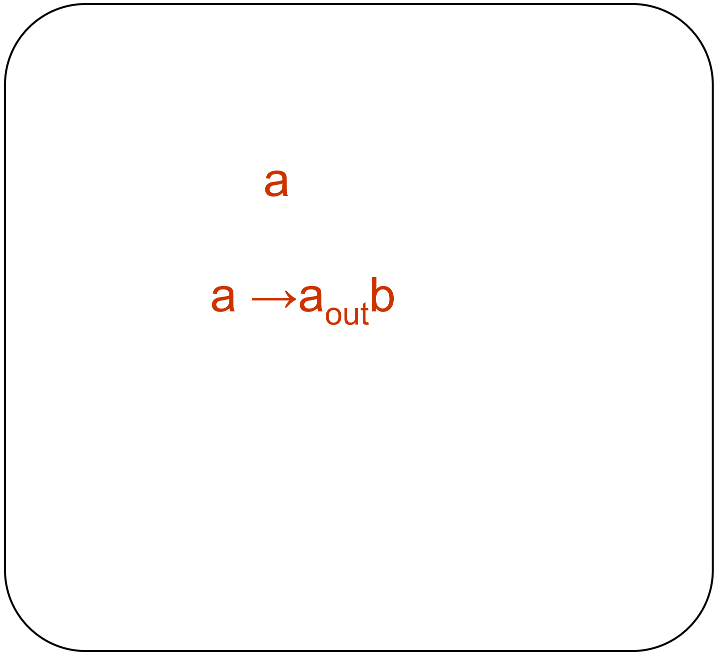


abb
b→b_out
aa

ab

$b \to b_{out}$

aab



aabb

b

$b \rightarrow b_{out}$

aabba

aabbab

Power of P systems

- Solving SAT problem:
  - Let us consider a propositional formula

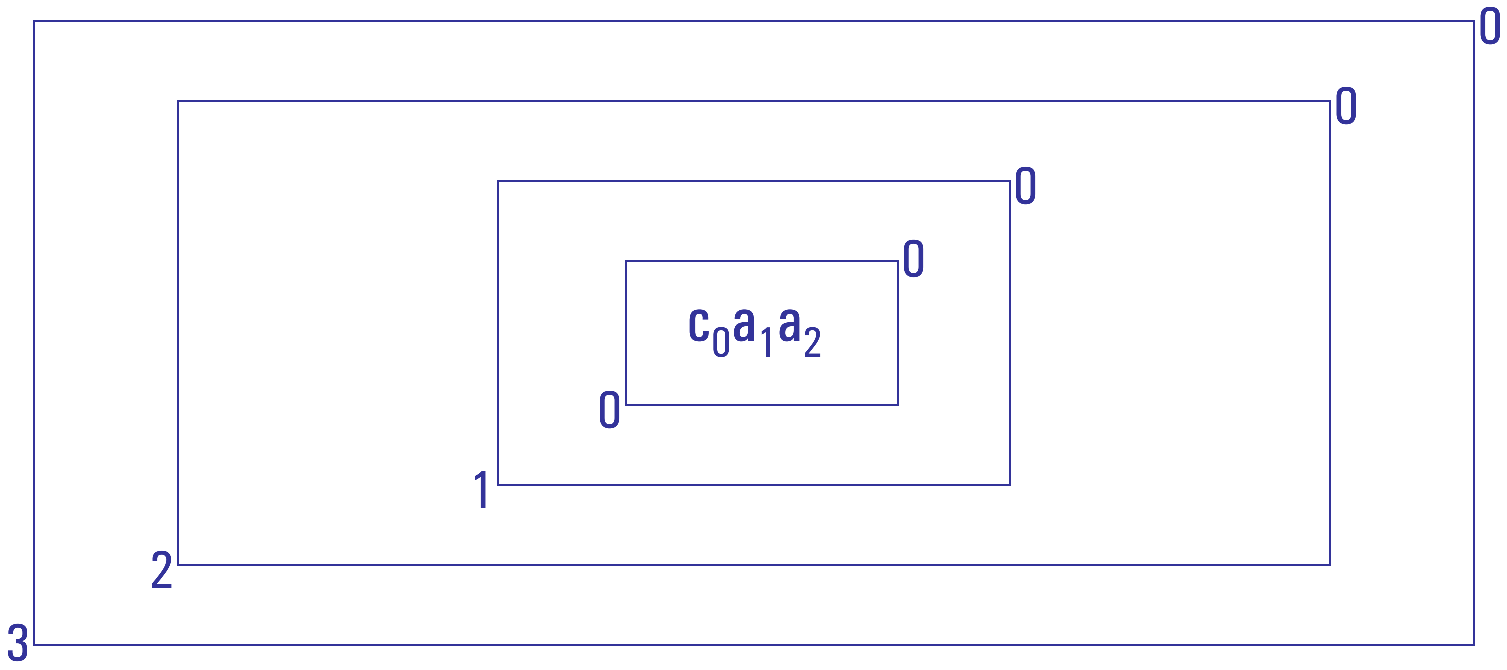
$$\gamma = C_1 \wedge C_2 \wedge \dots \wedge C_m,$$

with $$C_i = y_{i,1} \vee \dots \vee y_{i,p_i},$$

for some $m \geq 1$, $p_i \geq 1$, and $y_{i,j} \in \{x_k, \sim x_k \mid 1 \leq k \leq n\}$, for each $1 \leq l \leq m$, $1 \leq j \leq p_i$.

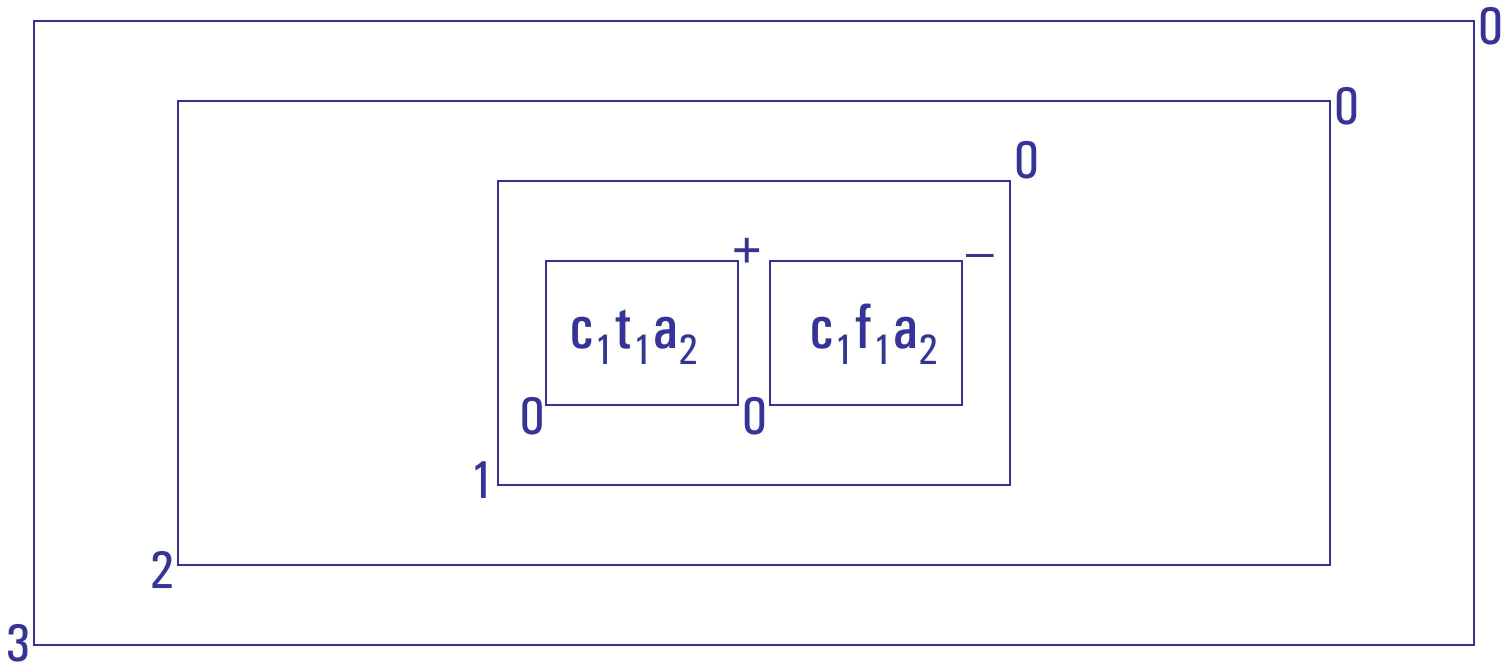
Power of P systems

- For example $\gamma = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$



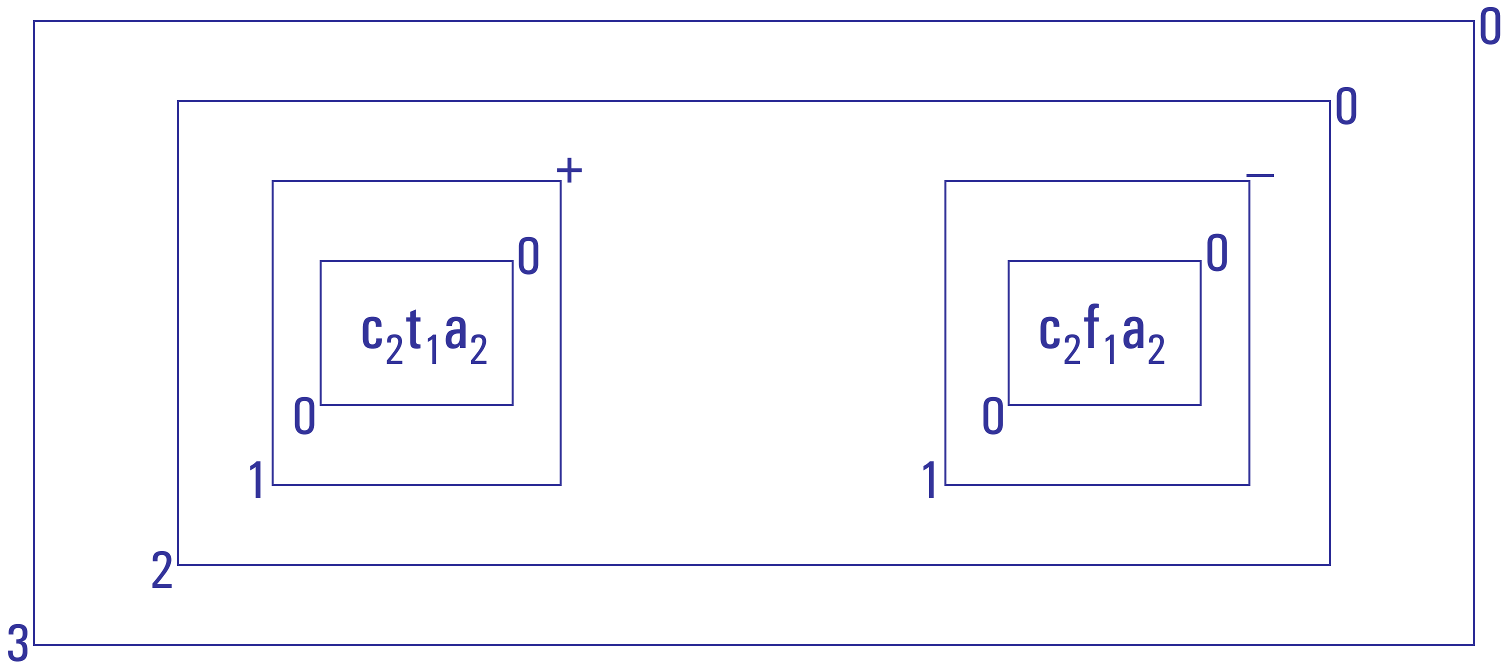
Power of P systems

- For example $\gamma = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$



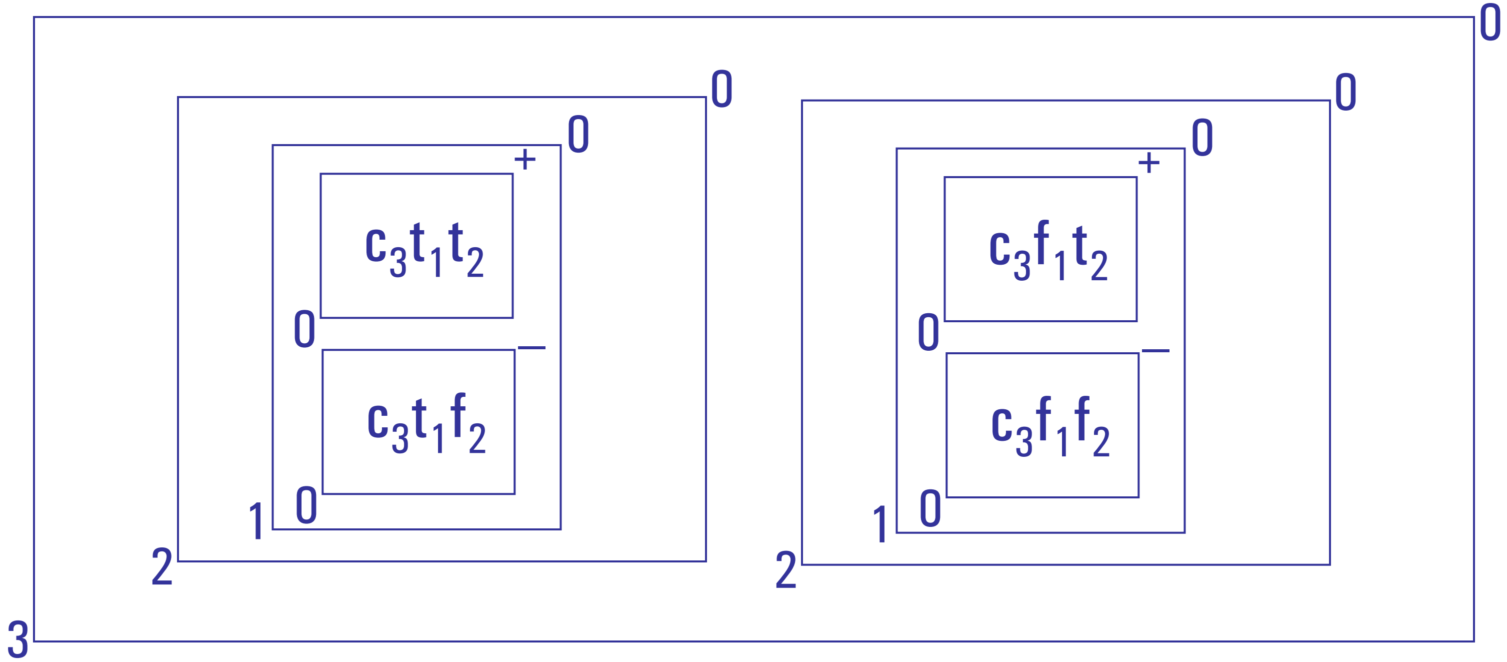
Power of P systems

- For example $\gamma = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$



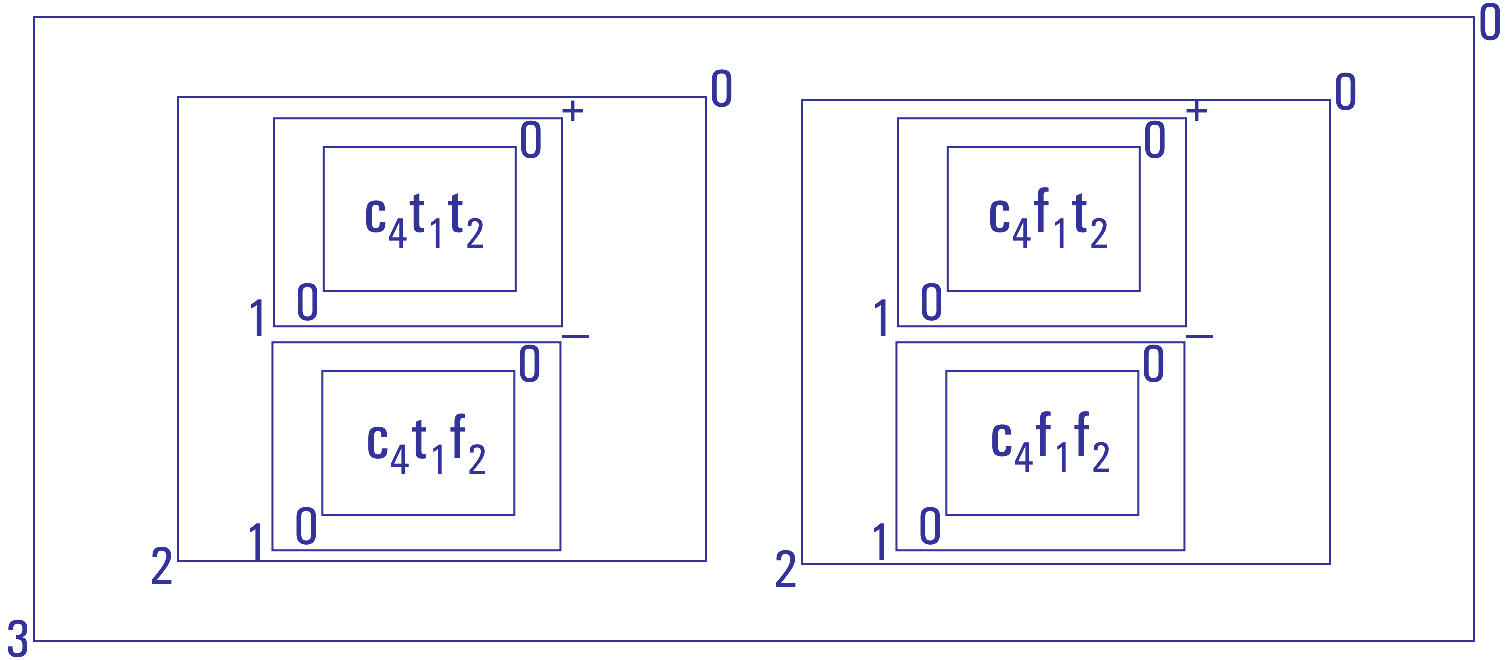
Power of P systems

- For example $\gamma = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$



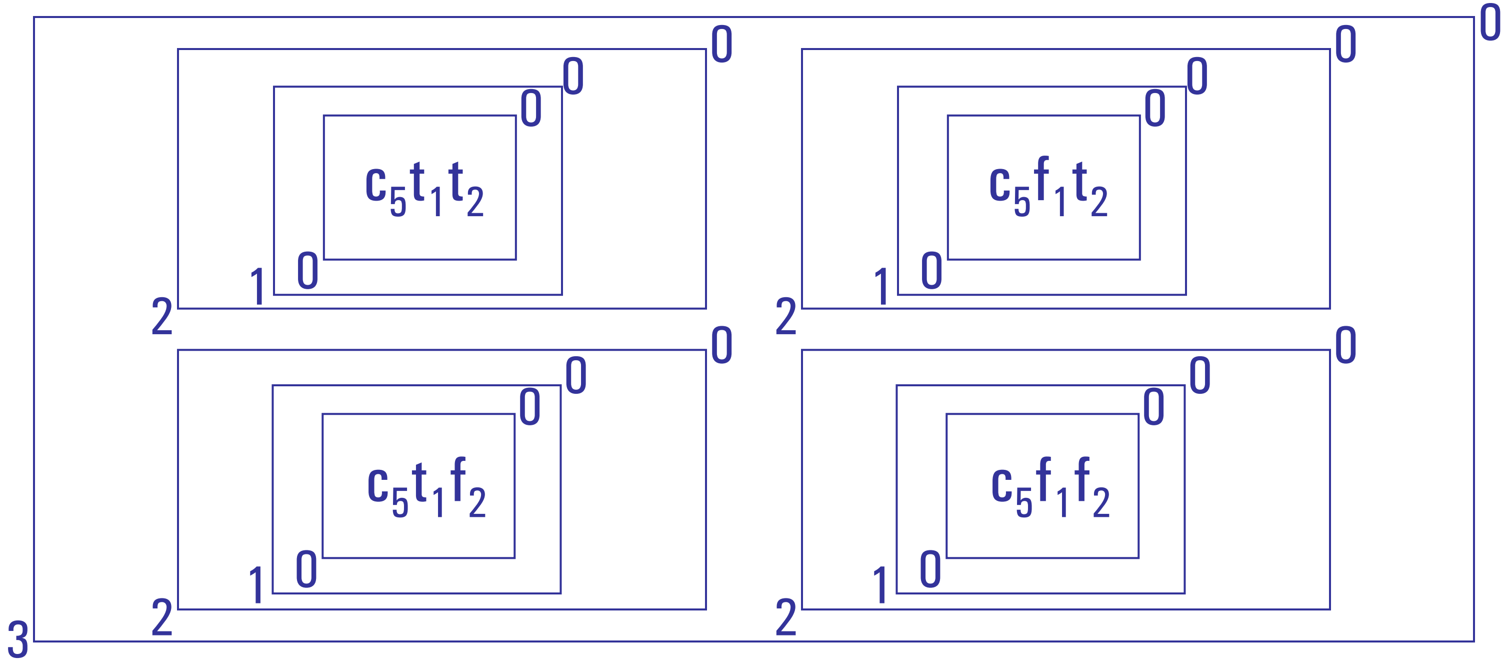
Power of P systems

- For example $\gamma = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$



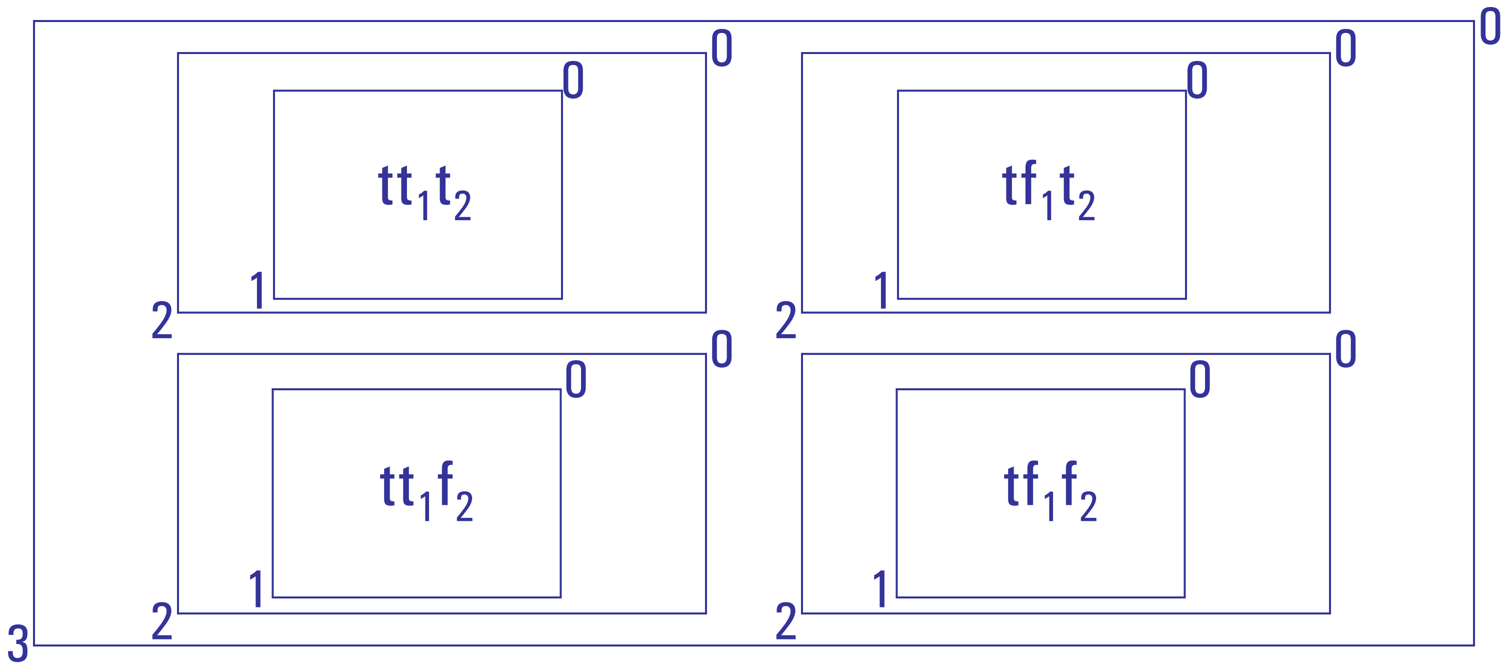
Power of P systems

- For example $\gamma = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$



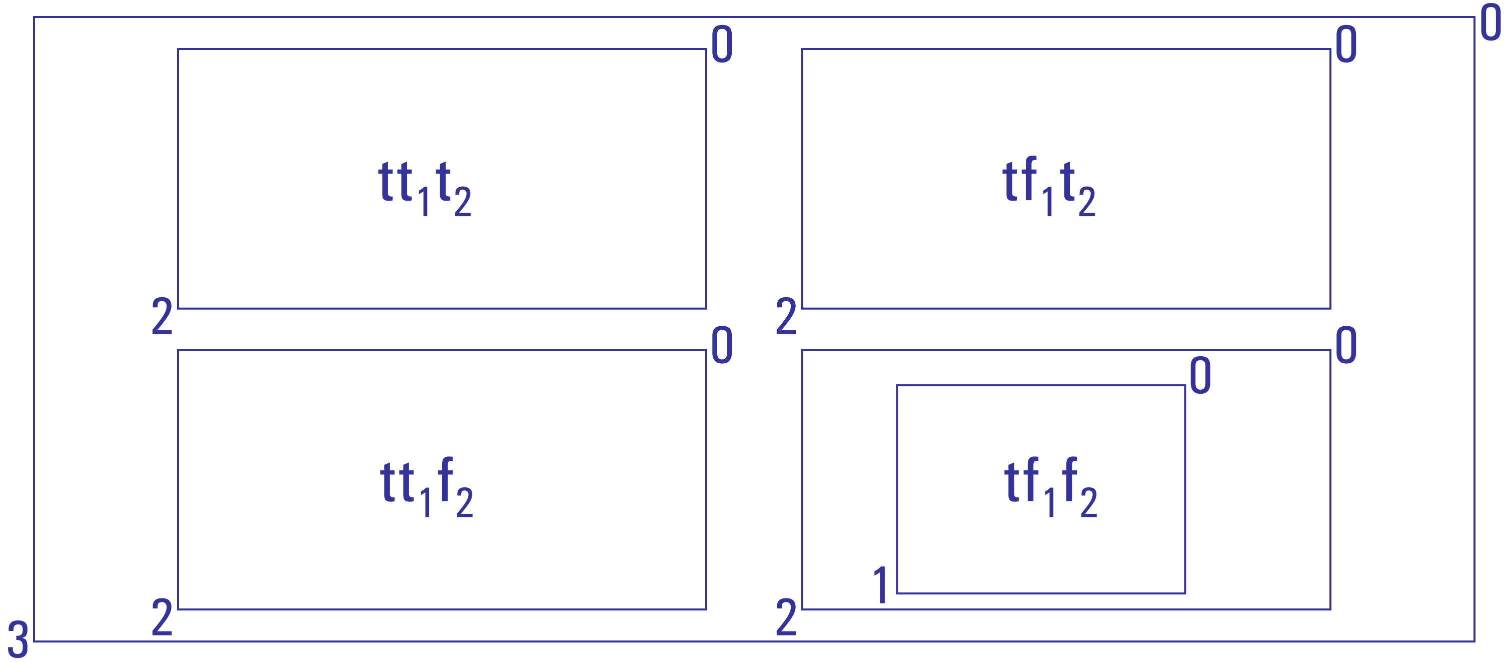
Power of P systems

- For example $\gamma = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$



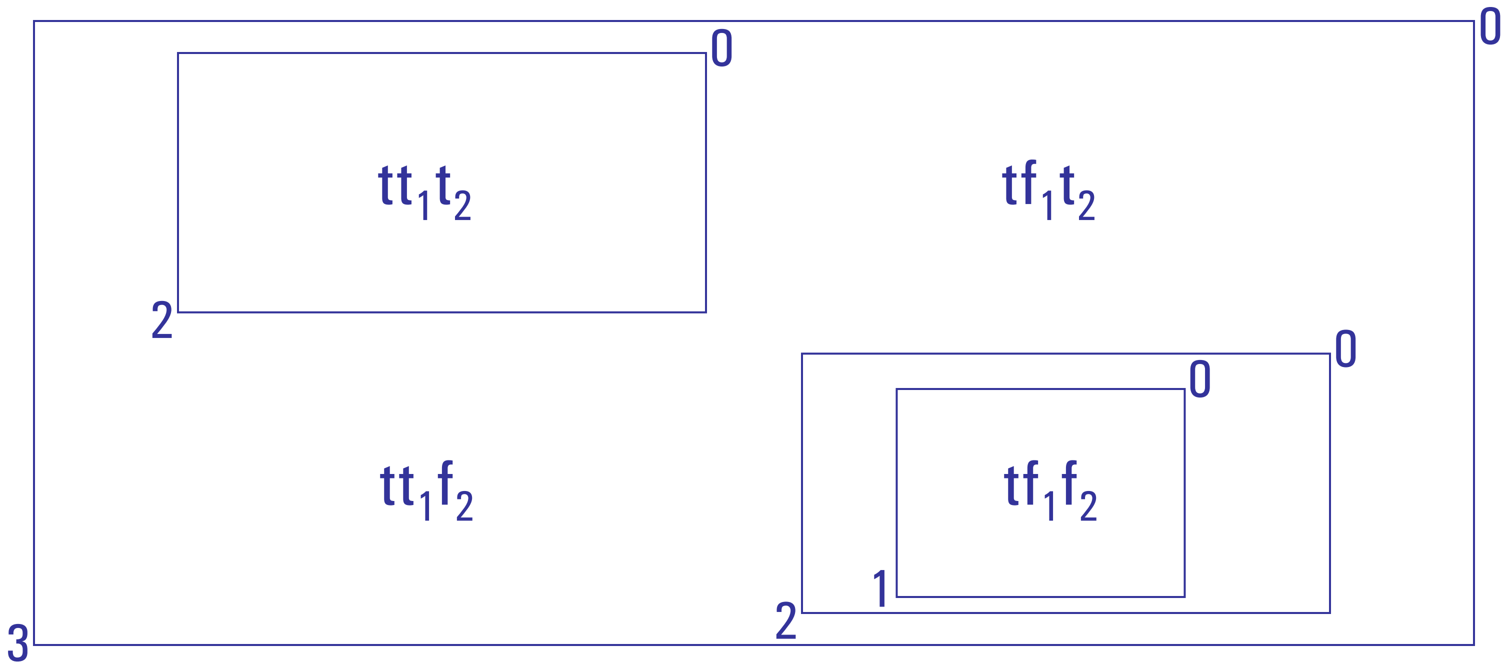
Power of P systems

- For example $\gamma = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$



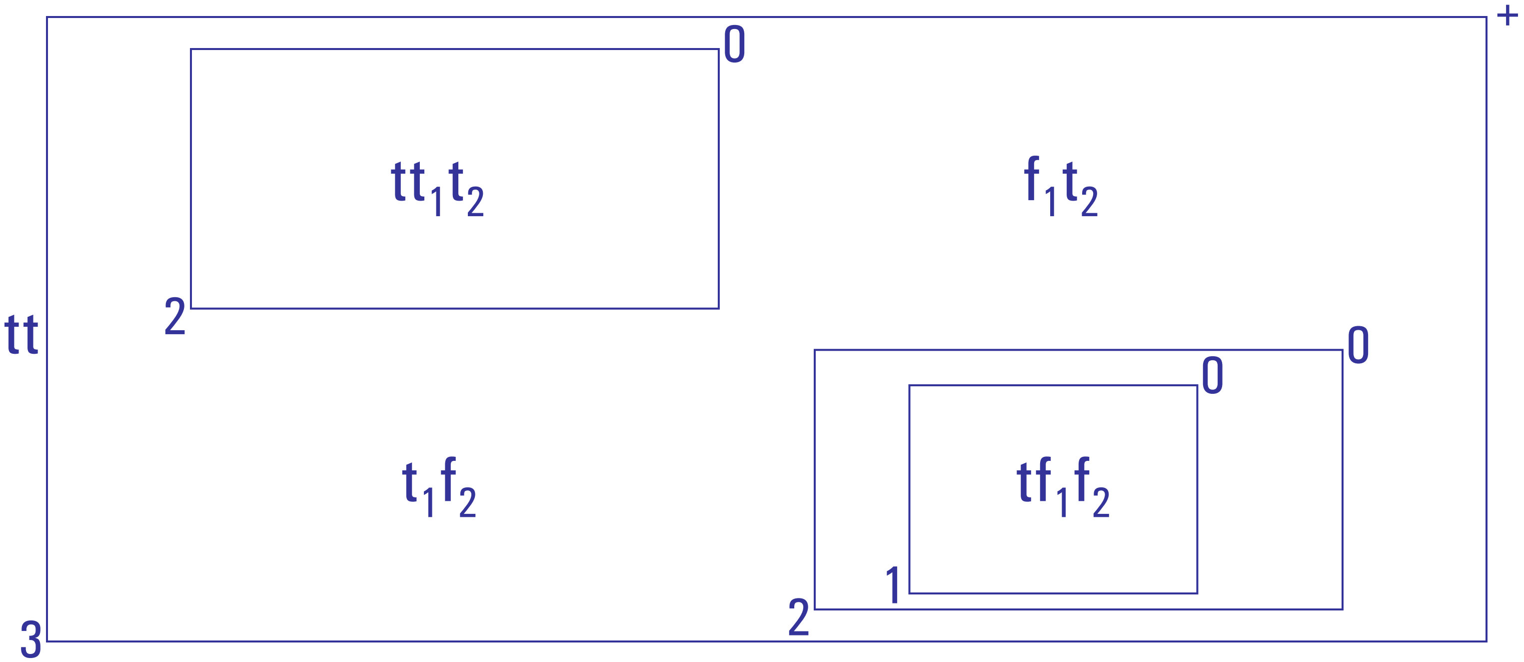
Power of P systems

- For example $\gamma = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$



Power of P systems

- For example $\gamma = (x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$



Other Aspects

- Additional features add any power?
- Generalized notion for normal forms
- Finding the descriptional complexity
- Contextual processing of string-objects
- Capturing the non-deterministic nature
- As accepting devices

Tissue P systems (or) Neural-like P systems

- Since cells live together and are associated with tissues and organs, inter-cellular communication becomes an essential feature.
- This communication is done through the protein channels established among the membranes of the neighboring cells.
- Tissue P systems (in short tP systems) are also motivated from the way neurons cooperate.

Biological Motivation

- A neuron has a body containing a nucleus; their membranes are prolonged by two classes of fibres: *dendrites, axon*
- Neurons process impulses in the complex net established by *synapses*.
- A synapse is a contact between an end bulb of a neuron and the dendrites of another neuron.
- The synthesis of an impulse and its transmission to adjacent neurons are done according to certain *states* of the neuron.

Definition: Neural-like P systems

- tP- systems is a construct

$\pi = (O, T, \sigma_1, \sigma_2, \ldots, \sigma_n, syn, i_{out})$ where

$syn \subseteq \{1,\ldots,m\} \times \{1,\ldots,m\}$.

$\sigma_i = (Q_i, s_{i,0}, w_{i,0}, R_i)$

R_i is a finite set of rules of the form $sx \rightarrow s'y$ or $sx \rightarrow s'(y, tar)$ where $s, s' \in Q_i$, $x, y \in O^*$, $tar \in \{go, out\}$.

Definition: Neural-like P systems

For $s, s' \in Q_i$, $x \in O^*$, $y \in O^*$, we write

$sx \Rightarrow_{min} s'y$ iff $sw \to s'w' \in P_i$, $w \subseteq x$, and $y = (x-w) \cup w'$;

$sx \Rightarrow_{par} s'y$ iff $sw \to s'w' \in P_i$, $w^k \subseteq x$, $w^{k+1} \not\subseteq x$, for some $k \geq 1$ and $y = (x-w^k) \cup w'^k$;

$sx \Rightarrow_{max} s'y$ iff $sw_1 \to s'w'_1, \ldots, sw_k \to s'w'_k \in P_i$, $k \geq 1$, such that $w_1 \ldots w_k \subseteq x$, $y = (x-w_1 \ldots w_k) \cup w'_1 \ldots w'_k$, and there is no $sw \to s'w' \in P_i$ such that

$w_1 \ldots w_k w \subseteq x$.

Transmitting the symbols

- Three transmitting modes are possible:

replication: each symbol a, for (a,go) appearing in w', is sent to each of the cells σ_j such that (i,j) $\in$ syn;

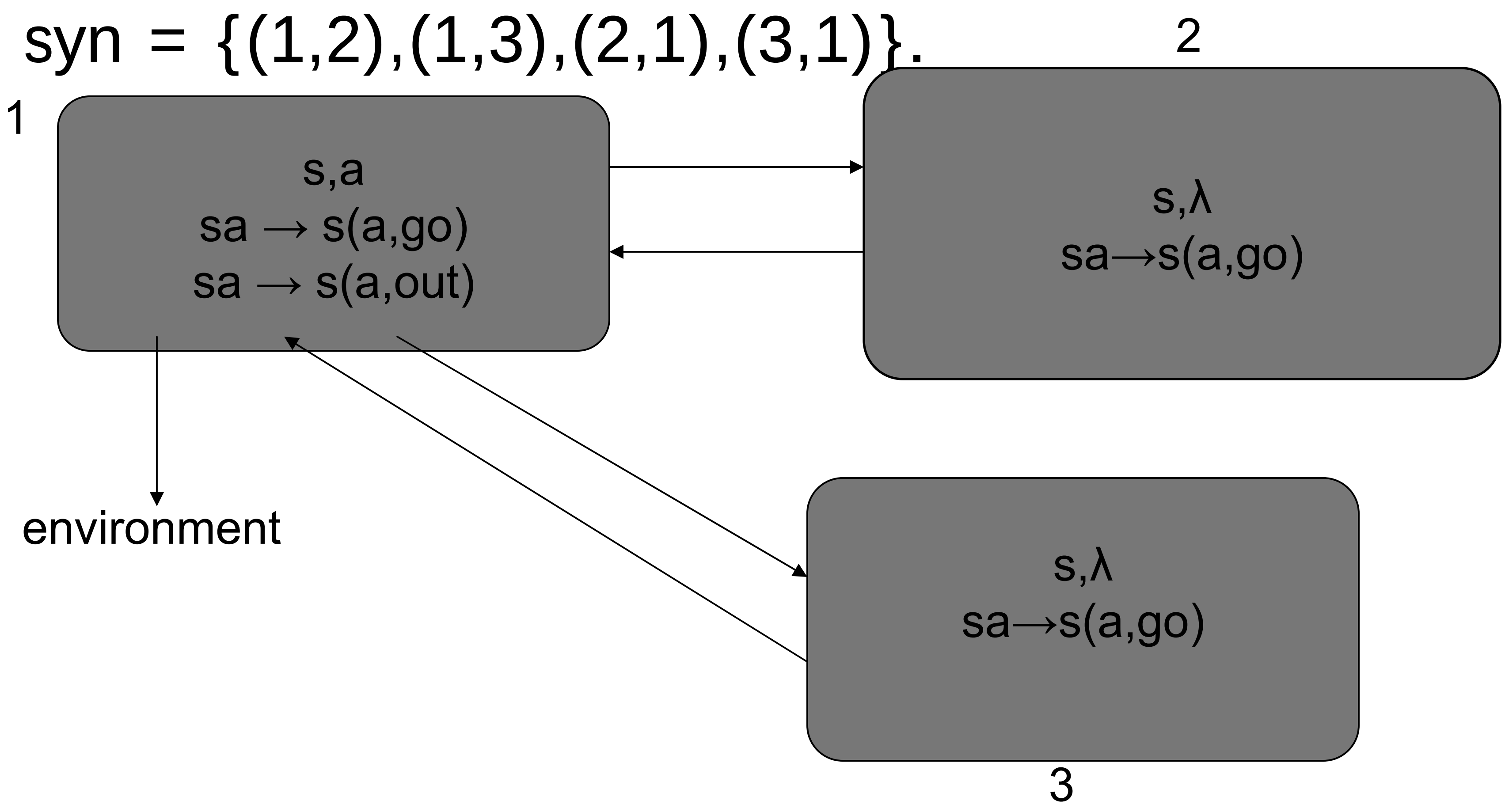
one: all symbols a appearing in w' in the form (a,go) are sent to one of the cells σ_j such that (i,j) $\in$ syn, nondeterministically chosen;

spread: the symbols a appearing in w' in the form (a,go) are nondeterministically distributed among the cells σ_j such that (i,j) $\in$ syn.

Example...

π = ({a}, σ_1 , σ_2 ,σ_3 , syn, i_{out}) where

syn = {(1,2),(1,3),(2,1),(3,1)}.



$N_{min,repl}(\pi_1) = \{ (n) \mid n \geq 1 \}$,

$N_{min,\beta}(\pi_1) = \{ (1) \}$, for $\beta \in \{one, spread\}$,

$N_{par,repl}(\pi_1) = \{ (2^n) \mid n \geq 0 \}$,

$N_{par,\beta}(\pi_1) = \{ (1) \}$, for $\beta \in \{one, spread\}$,

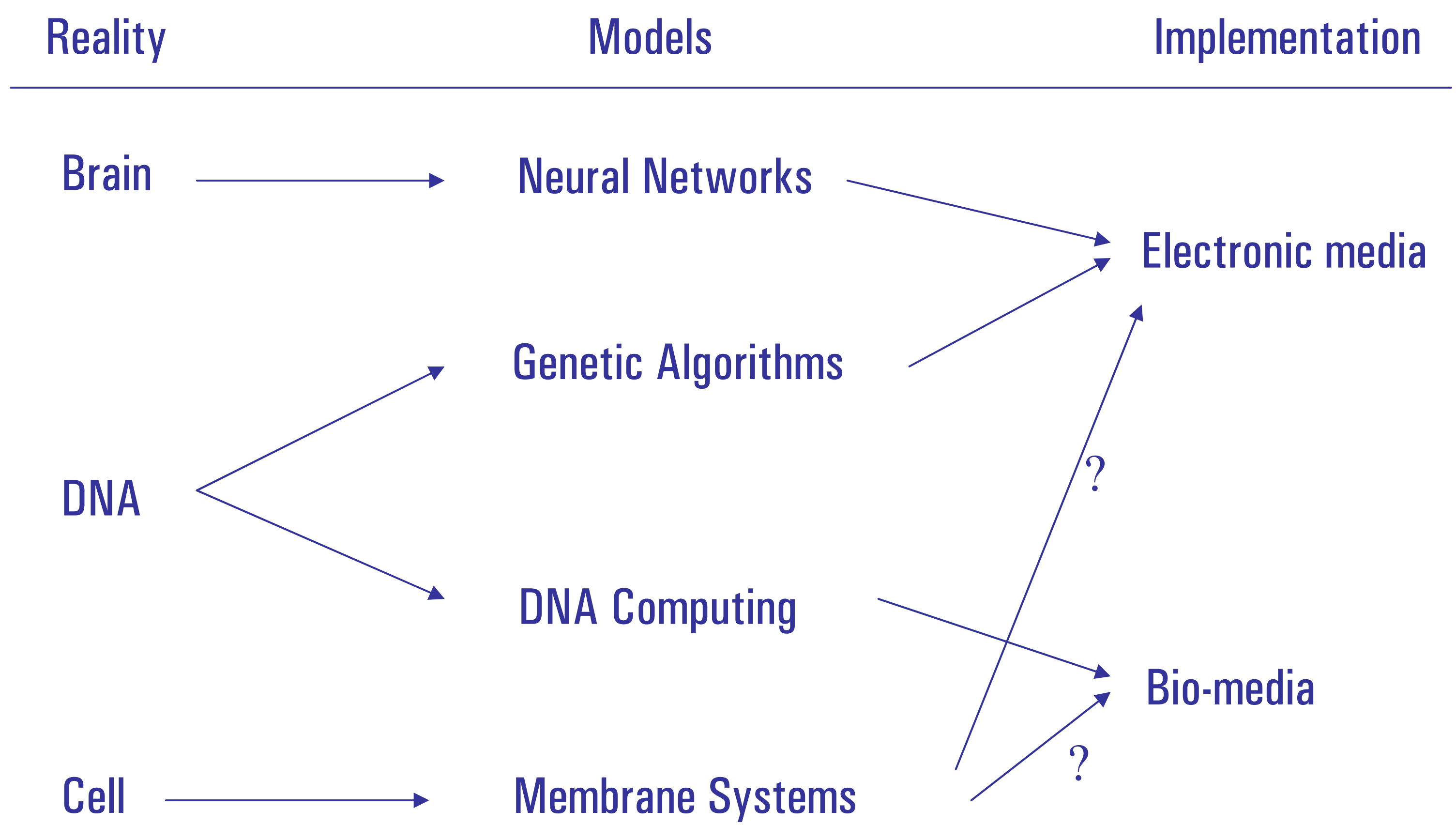
$N_{max, repl}(\pi_1) = \{ (n) \mid n \geq 1 \}$,

$N_{max,\beta}(\pi_1) = \{ (1) \}$, for $\beta \in \{one, spread\}$.

Our Contribution

- Membrane creation
- Message carriers
- Generalized normal form
- Descriptional complexities
- Contextual P systems
- Hybrid P systems
- Probabilistic P systems
- P automata

Future Direction ...



REFRENCES

1) C.S. Calude and Gh. Păun. Computing with Cells and Atoms. Taylor and Francis, London, 2001

2) Gh. Păun. Membrane Computing. An introduction, Springer-Verlag, Berlin, 2002

3) http://psystems.disco.unimib.it

Thank You