Brainstorming Week on Membrane Computing (BWMC 2025)
22-24 January 2025, Sevilla Spain

Computing by plasmids

José M. Sempere
Valencian Research Institute for Artificial Intelligence (VRAIN)
Valencian Graduate School and Research Network of AI (VALGRAI)
Department of Information Systems and Computation
Universitat Politècnica de València

jsempere@dsic.upv.es

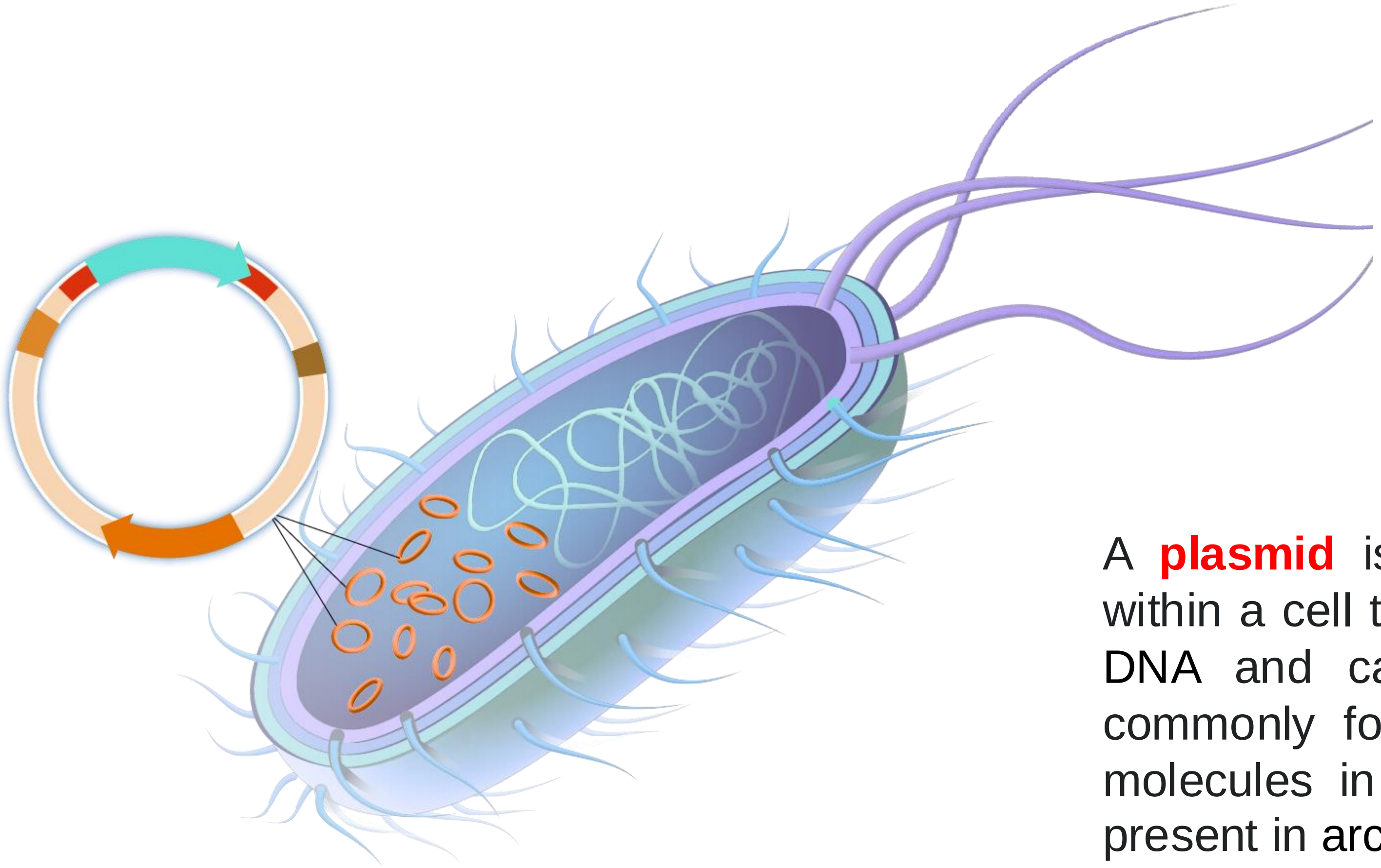
jsempere.webs.upv.es

UPV VRAIN

valgrAI

DSIC DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA



A **plasmid** is a small, extrachromosomal DNA molecule within a cell that is physically separated from chromosomal DNA and can replicate independently. They are most commonly found as small circular, double-stranded DNA molecules in bacteria; however, plasmids are sometimes present in archaea and eukaryotic organisms.

(Wikipedia)

Use plasmids to achieve movile/distributed computing and reduce the descriptional complexity of P systems

Related works

F.G.C. Cabarle, X. Zeng, N. Murphy, T. Song, A. Rodríguez-Patón, X. Liu. Neural-like P systems with plasmids. Information and Computation, Vol. 281, 194776. (2021)

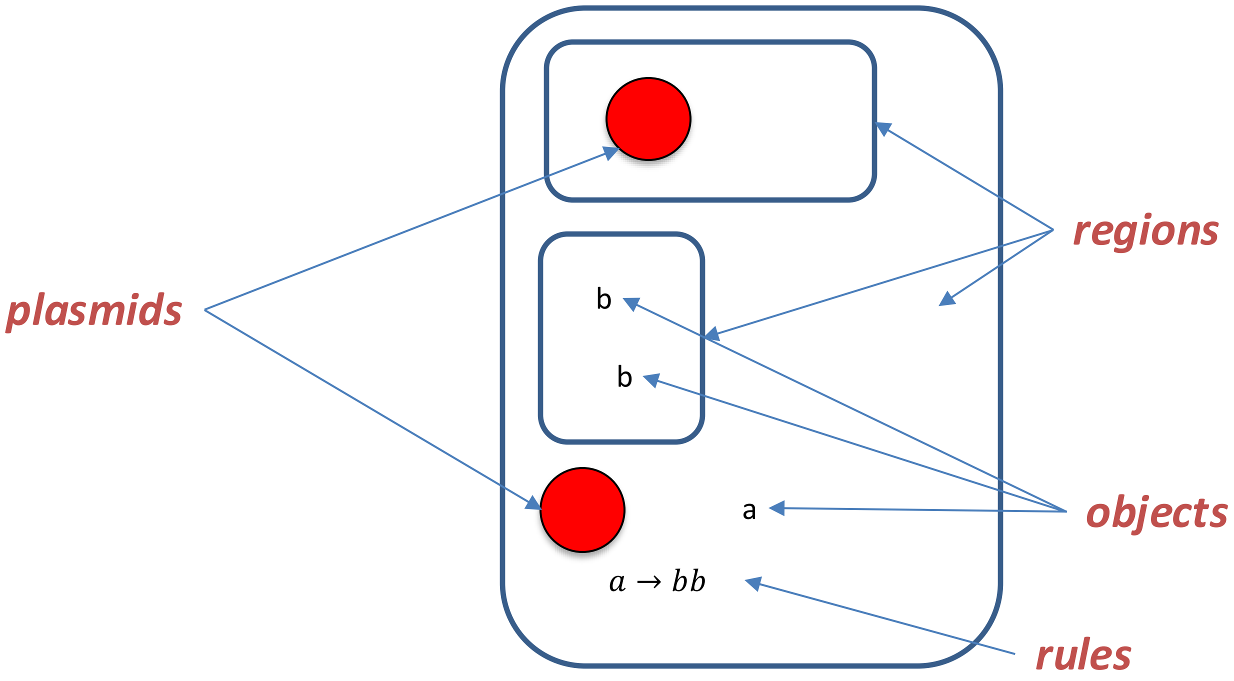
The authors introduce plasmid objects that have no explicit rules in the SP N framework

Y. Li, B. Song and X. Zeng. Neural-Like P Systems With Plasmids and Multiple Channels. IEEE Transactions on NanoBioscience, vol. 22, no. 2, (2023) 420-429

The authors combine plasmid objects that have no explicit rules and multiple channels in the SP N framework

What is a plasmid in the membrane computing framework ?

A **plasmid** is a finite set of rules (with priorities) that is surrounded by a membrane.



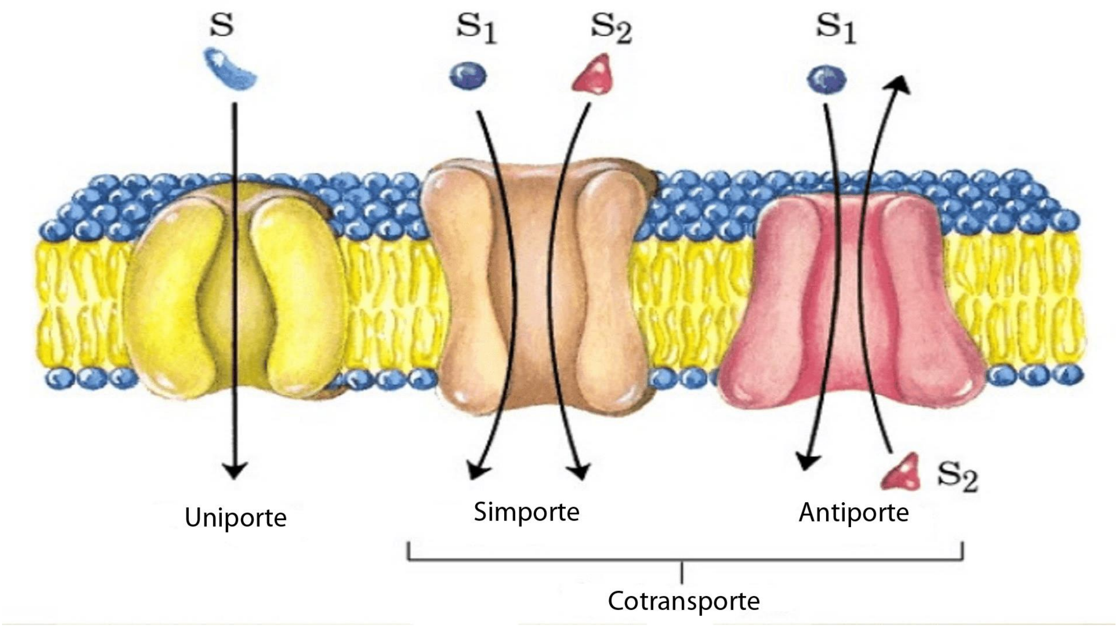
What operations can be made with plasmids ?

Plasmid mobility

'in symport': $p_i\, t[_k\, a]_k \rightarrow r[_k\, p_i\, b]_k$

'out symport': $t[_k\, p_i\, a]_k \rightarrow p_i\, r[_k\, b]_k$

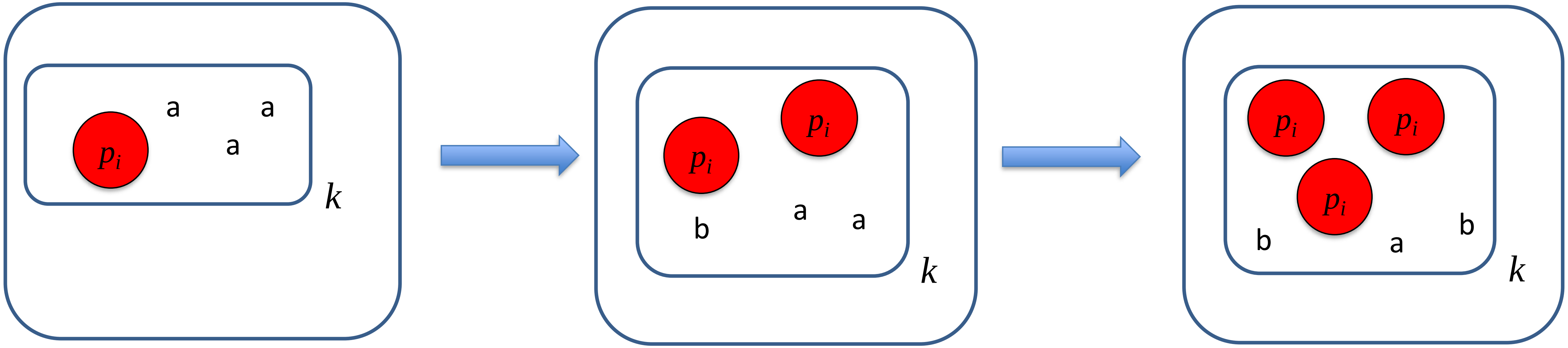
'anty-port': $p_i\, t[_k\, p_j\, a]_k \rightarrow p_j\, r[_k\, p_i\, b]_k$



What operations can be made with plasmids ?

Plasmid replication (initially, minimal paralellism occurrence, only one copy is produced)

$$[\, p_i \mathrm{a}]_k \rightarrow [p_i p_i b]_k$$



A P system with q plasmids of degree $m \geq 1$ (Ppl_m^q) is defined by

$$\Pi = (V, H, \mu, w_1, w_2, \dots, w_m, R_o, R_p, \sigma, z_1, z_2, \dots, z_m, z_\infty, p_1, p_2, \dots, p_q, i_0)$$, where:

1) V is an alphabet of objects.
2) H is an alphabet of membrane labels.
3) μ is a membrane estructure.
4) $\mathrm{w}_1, \mathrm{w}_2, \dots, \mathrm{w}_\mathrm{m}$ are the initial multisets of objects at every region.
5) R_o is a set of rules for objects and membranes.
6) R_p is a set of rules for plasmid mobility and plasmid replication.
7) σ is a partial order over R_0 and R_p.
8) $z_1, z_2, \dots, z_m, z_\infty$ are the initial multisets of plasmids at every region and the environment.
9) $p_1, p_2, \dots, p_q$ are plasmids, where every plasmid is defined by a finite set of rules for objects.
10) $i_0 \in \{1, \dots, m\} \cup \infty$ is the output membrane.

Remarks

- Initially plasmid rules only affect to objects (they do not affect to plasmids or membranes)
- Initially we only allow plasmid mobility and replication (no plasmid destruction)
- Plasmid replication is applied with minimal parallelism occurrence (only applied to one copy).

An example

$P_1 : \{a \rightarrow a_{out}\}$

$P_2 : \{ab \rightarrow c_{out}\}$

$R_o : [\, a\, [\ ]_3\,]_1 \rightarrow [\ [\, a\,]_3\,]_1$

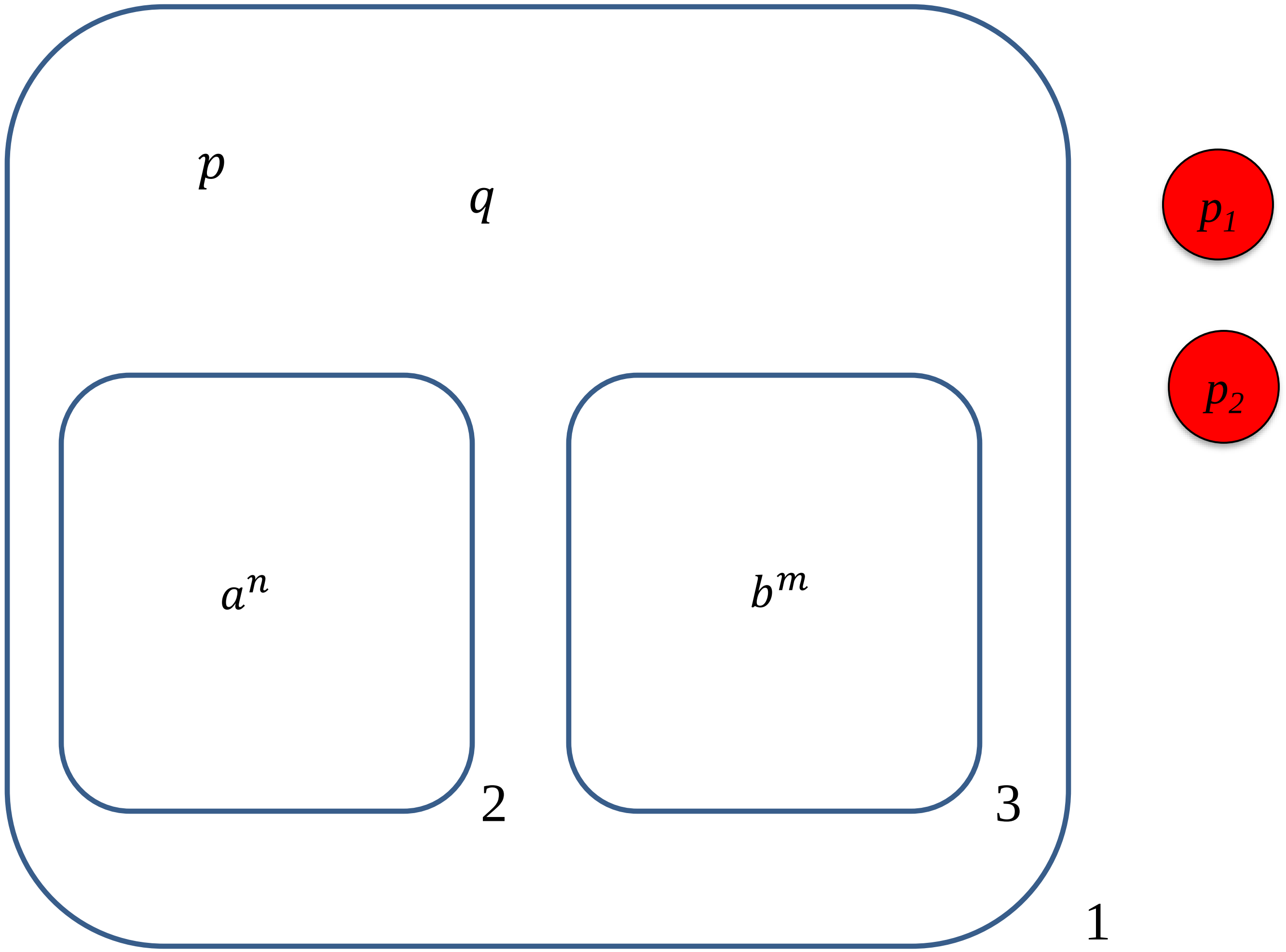
$R_p :$

$P_1 [\, p\,]_1 \rightarrow [\, P_1\,]_1$

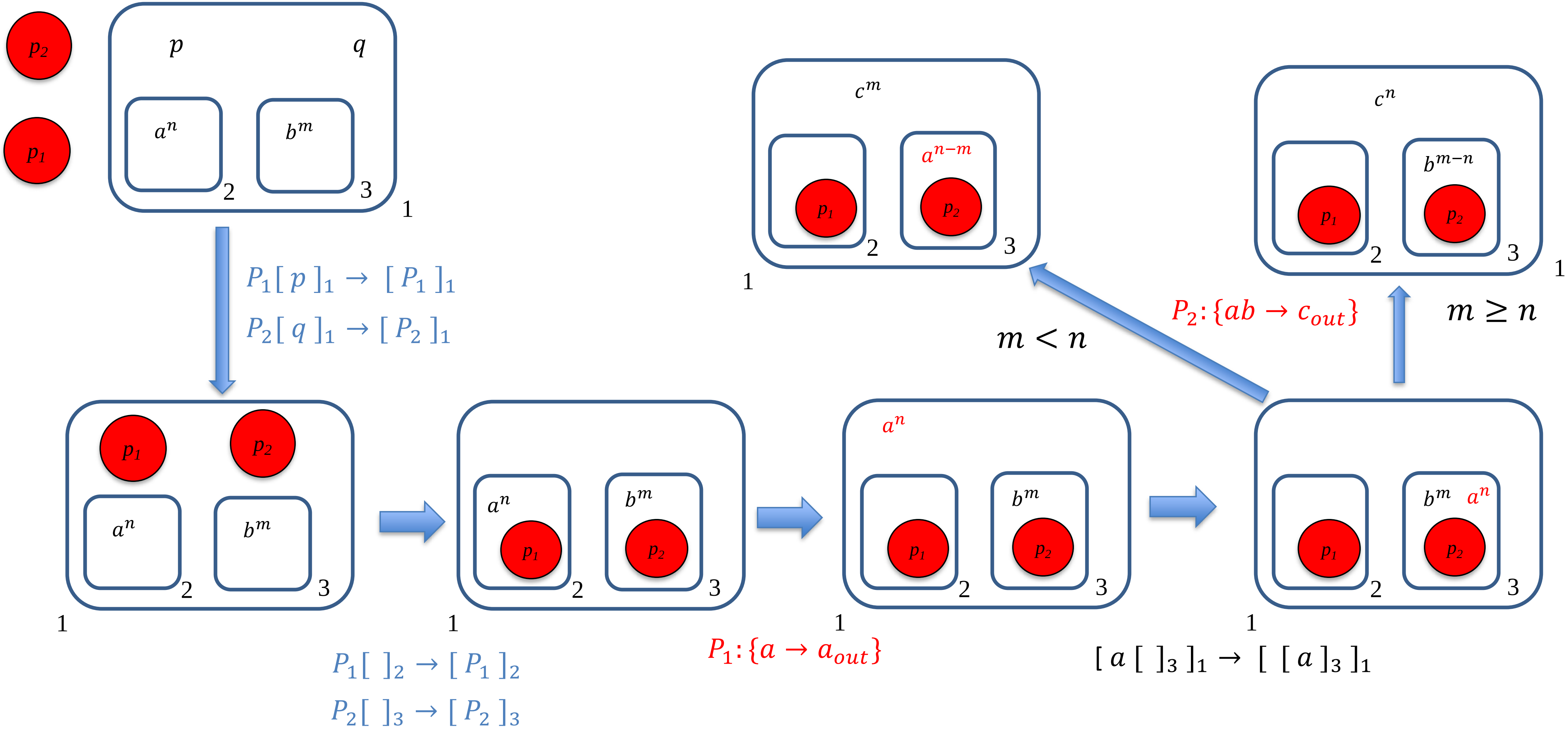
$P_2 [\, q\,]_1 \rightarrow [\, P_2\,]_1$

$P_1 [\ ]_2 \rightarrow [\, P_1\,]_2$

$P_2 [\ ]_3 \rightarrow [\, P_2\,]_3$



An example


p_2
p_1
p
q
a^n
b^m
2
3
1
$P_1[\ p\ ]_1 \rightarrow [\ P_1\ ]_1$
$P_2[\ q\ ]_1 \rightarrow [\ P_2\ ]_1$
$P_1[\ ]_2 \rightarrow [\ P_1\ ]_2$
$P_2[\ ]_3 \rightarrow [\ P_2\ ]_3$
$P_1 : \{a \rightarrow a_{out}\}$
$[\ a\ [\ ]_3\ ]_1 \rightarrow [\ [\ a\ ]_3\ ]_1$
$P_2 : \{ab \rightarrow c_{out}\}$
$m < n$
$m \geq n$
c^m
a^{n-m}
c^n
b^{m-n}

Universality

A natural way to simulate register machines:

- Every basic instruction is a plasmid
- Every register is a membrane inside the skin membrane
- The program runs through plasmid mobility
- The instruction counter is defined by a specific set of objects related to the register machine instruction labels.

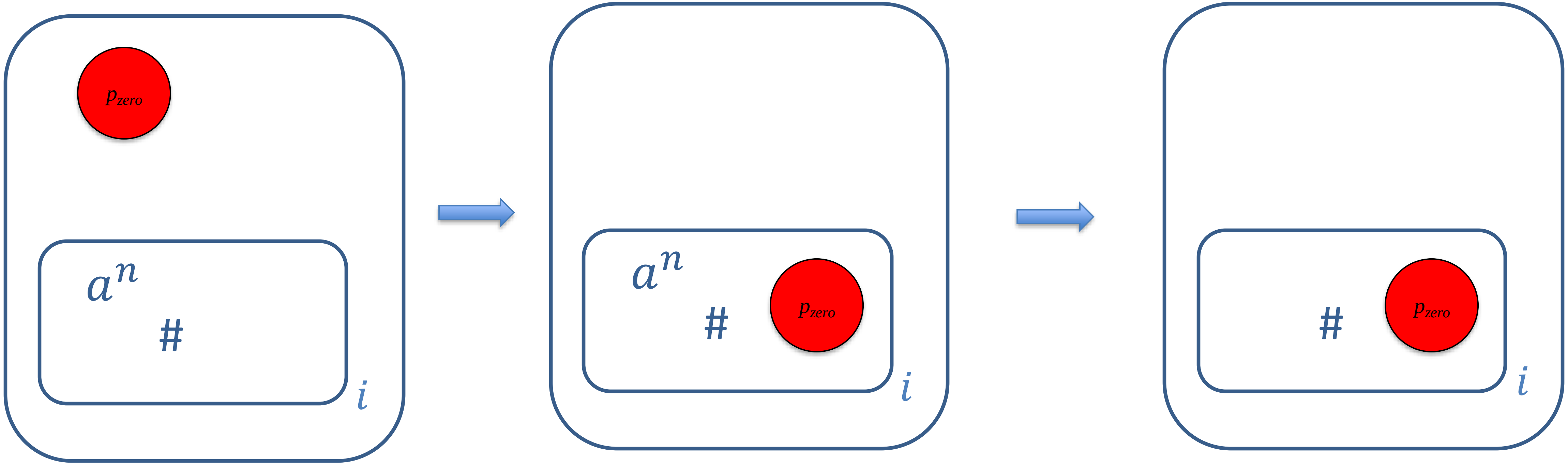
Let us suppose that we encode $n \in \mathbb{N}$ as $a^n\#$

$zero(i)$

$[\, i \,] \leftarrow 0$

$P_{zero}[\ ]_i \rightarrow [\, P_{zero}\,]_i$

$P_{zero} = \{a \rightarrow \lambda\}$

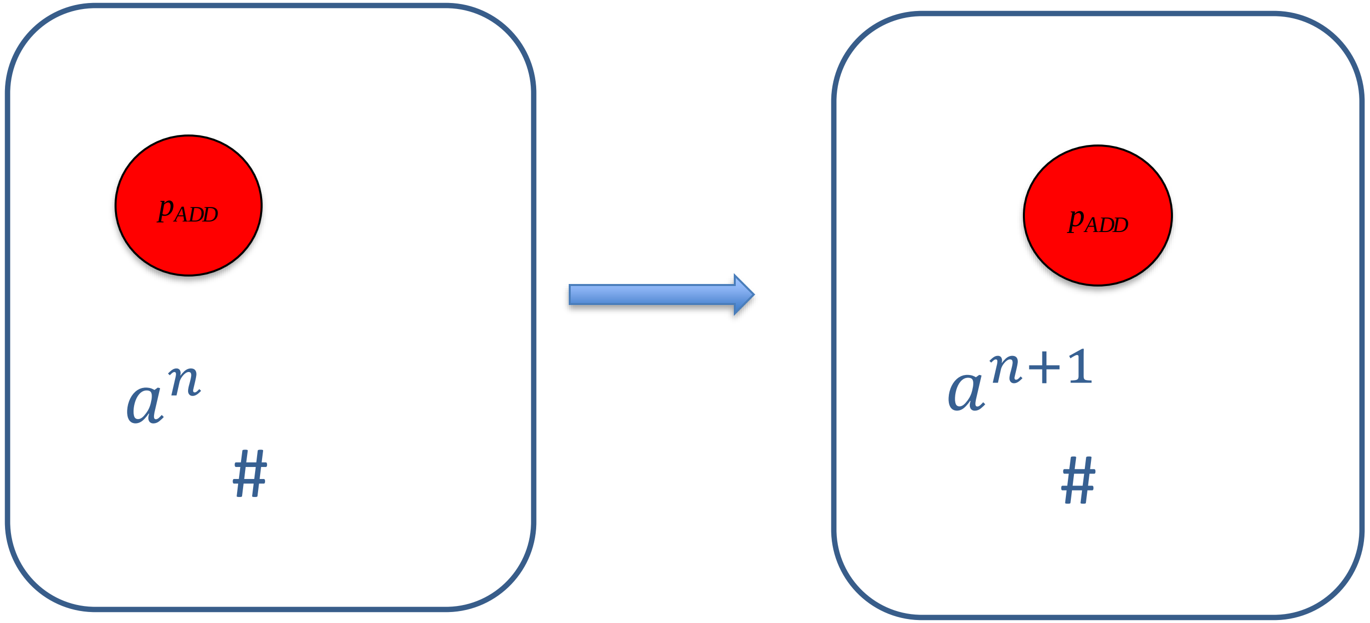


Let us suppose that we encode $n \in \mathbb{N}$ as $a^n\#$

$\mathrm{ADD}(i, l_1, l_2)$

$[\, i \,] \leftarrow [i] + 1$

$$P_{ADD} = \{a\# \rightarrow aa\#\}$$

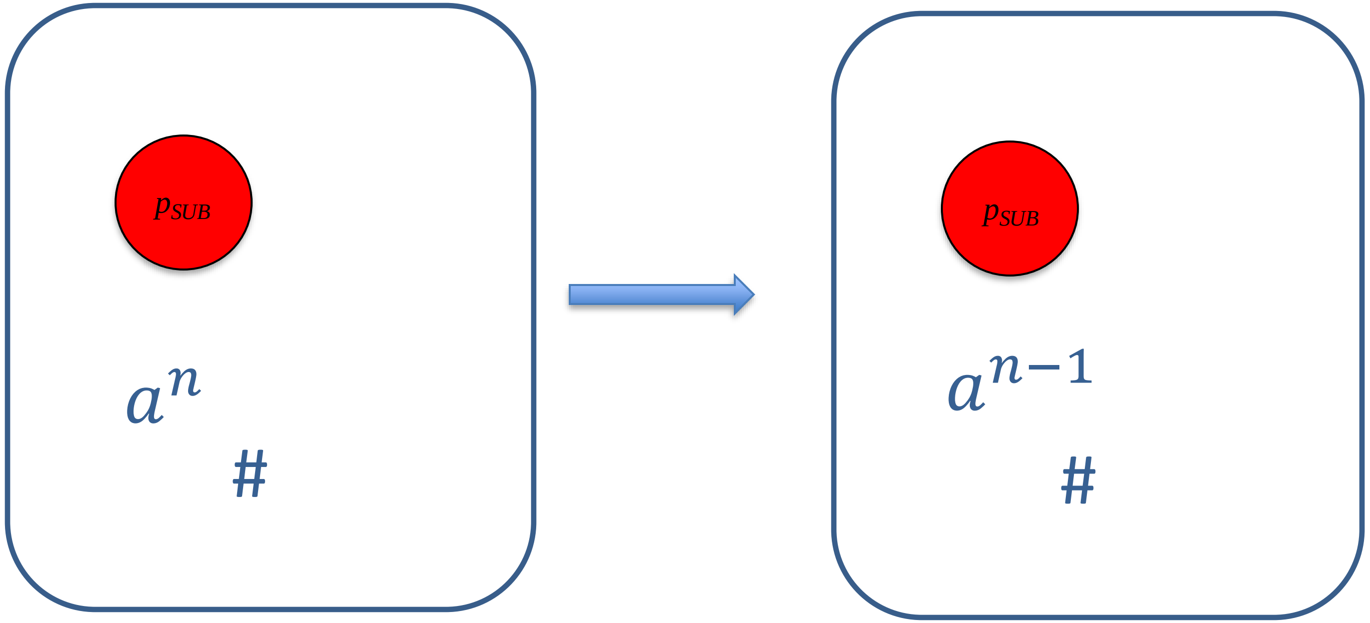


Let us suppose that we encode $n \in \mathbb{N}$ as $a^n\#$

$\mathrm{SUB}(i, l_1, l_2)$

$[\,i\,] \leftarrow [i] - 1$

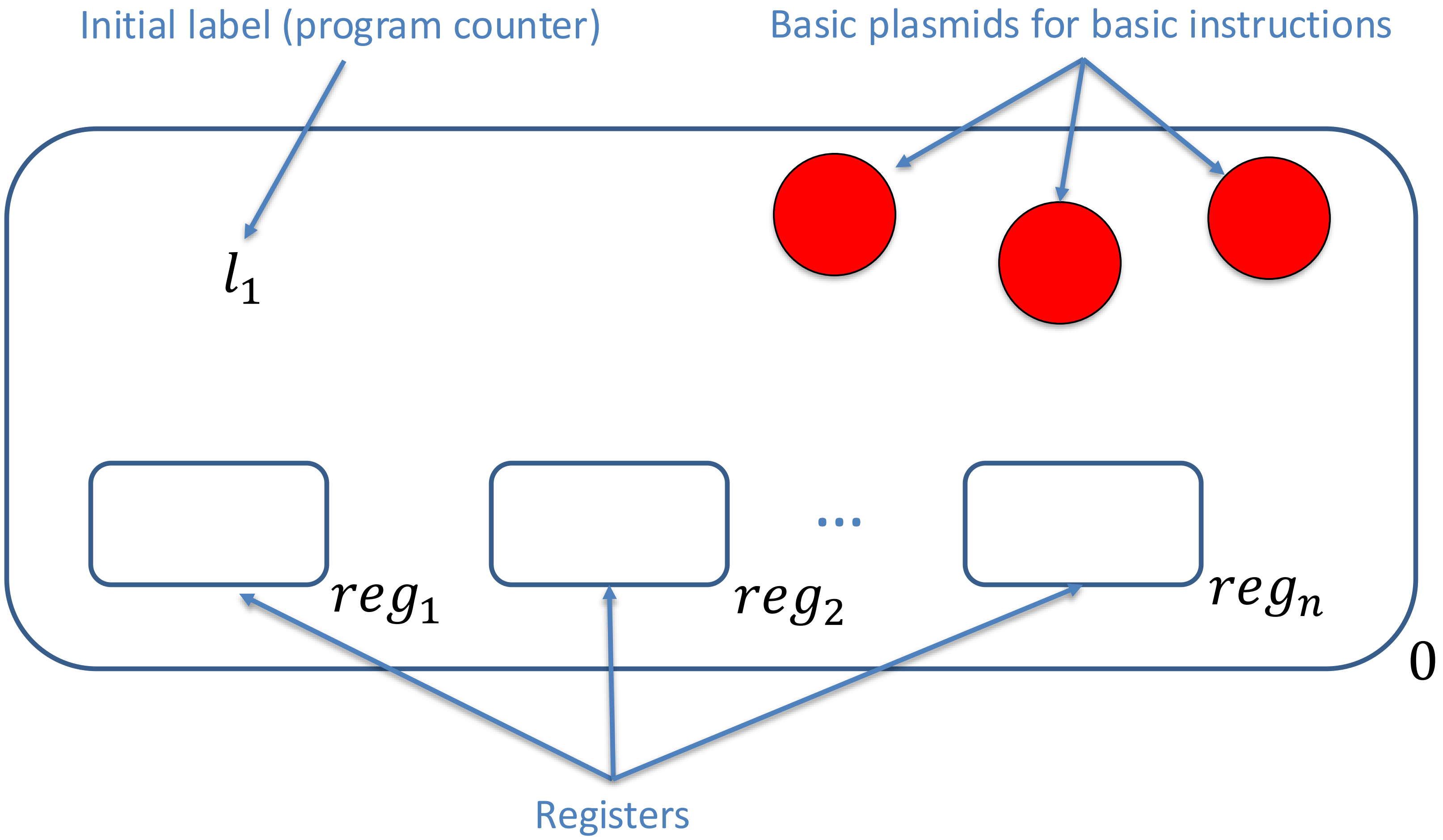
$P_{SUB} = \{a\# \rightarrow \#\}$



$n > 0$

The execution of a program with n registers

l_1: instruction1
l_2: instruction2
...
l_p: instruction$p$



The execution of a program with n registers

l_1:ADD(i, l_2, l_3)

Execution init

$[l_1 P_{ADD} [\ ]_i]_0 \to [\,[\,l_1 P_{ADD}]_i]_0$

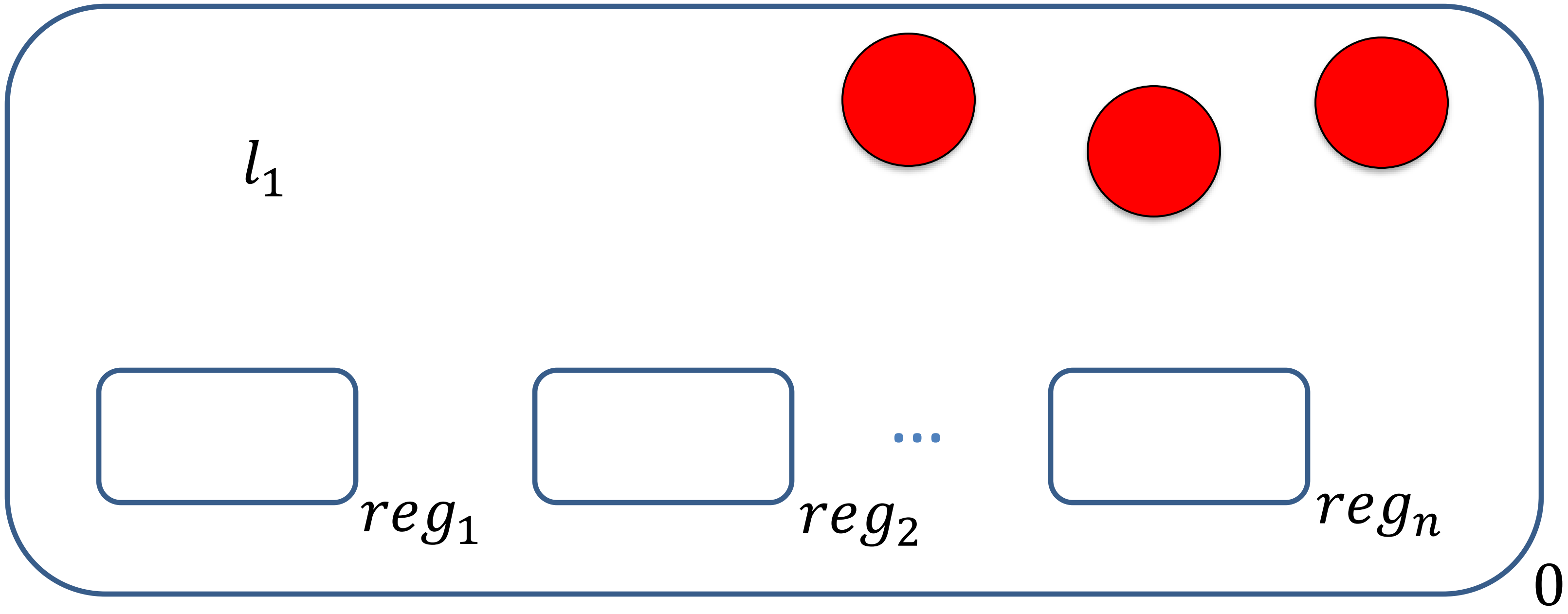
Calculation

$P_{ADD} = \{a\# \to aa\#\}$

Non-deterministic next instruction addressing

$[\,[\,l_1 P_{ADD}\,]_i\,]_0 \to [\,l_2 P_{ADD}\,[\,]_i]_0$

$[\,[\,l_1 P_{ADD}\,]_i\,]_0 \to [\,l_3 P_{ADD}\,[\,]_i]_0$



Work in progress and future research

- Connection with other models: initially consider P colonies
- Self-replicated machines: plasmids inside themselves
- Plasmid evolution: polymorphic rules
- Plasmid mobility: introducing polarizations, nested membranes, …
- Plasmid degradation: allow the dissolution of the plasmid membrane (so, the plasmid rules remain in the host region and the plasmid dissapears)

Brainstorming Week on Membrane Computing (BWMC 2025)
22-24 January 2025, Sevilla Spain

Thank you !!

jsempere@dsic.upv.es

jsempere.webs.upv.es