introducing

Membrane Computing

Models of Computation

Utrecht 7 june 2012

Hendrik Jan Hoogeboom

LIACS / Computer Science Leiden

Leiden Center for Natural Computing

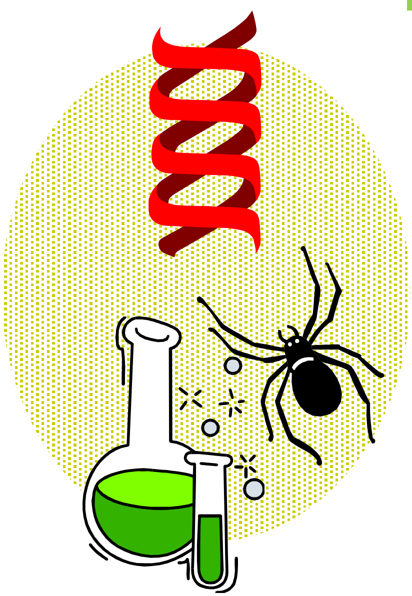
natural computation

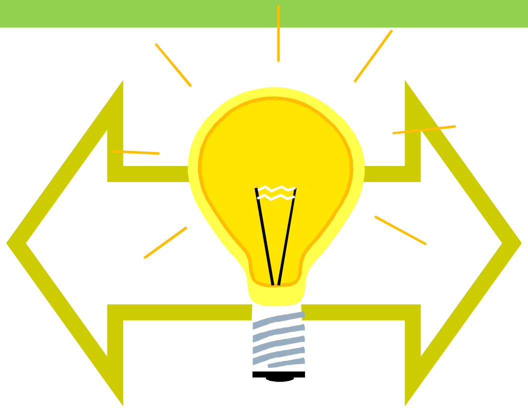
dna sorting &
protein networks

understanding nature
as a computational process

bio inspired computing

neural netw &
genetic alg





bio hardware

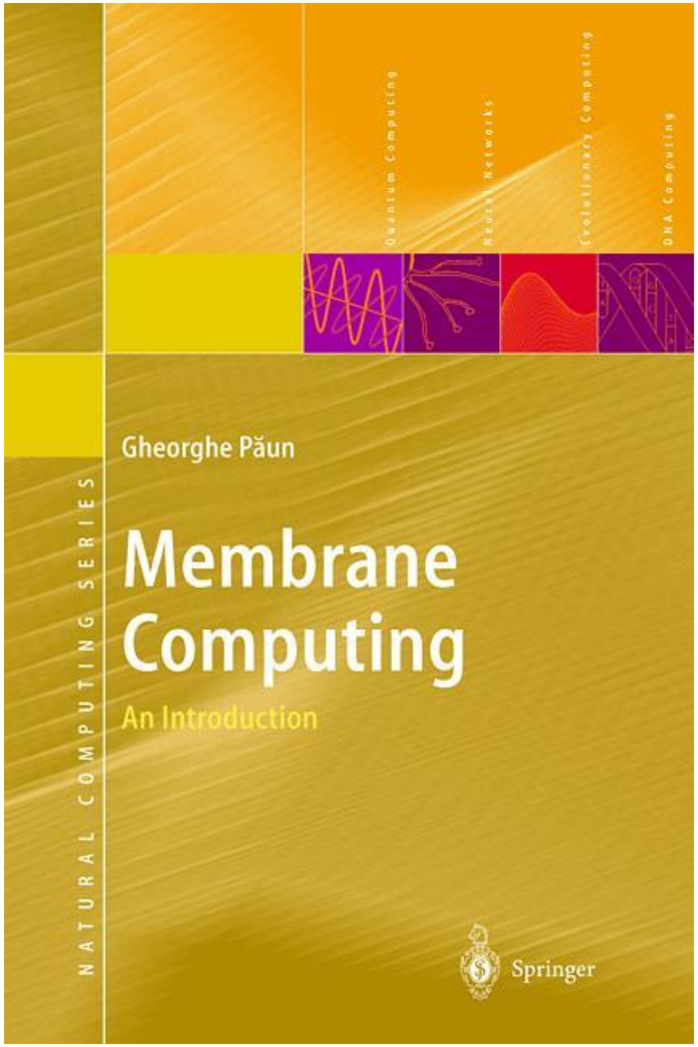
DNA computing

bio-informatics

gene finding

Membrane Computing

introducing some features
(almost) no formalisms
no (real) proofs



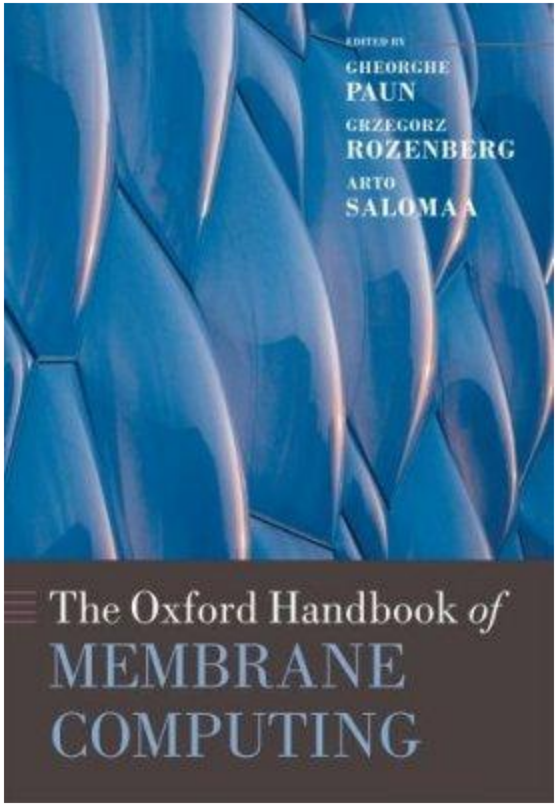
Gheorghe Păun

Membrane Computing
An Introduction

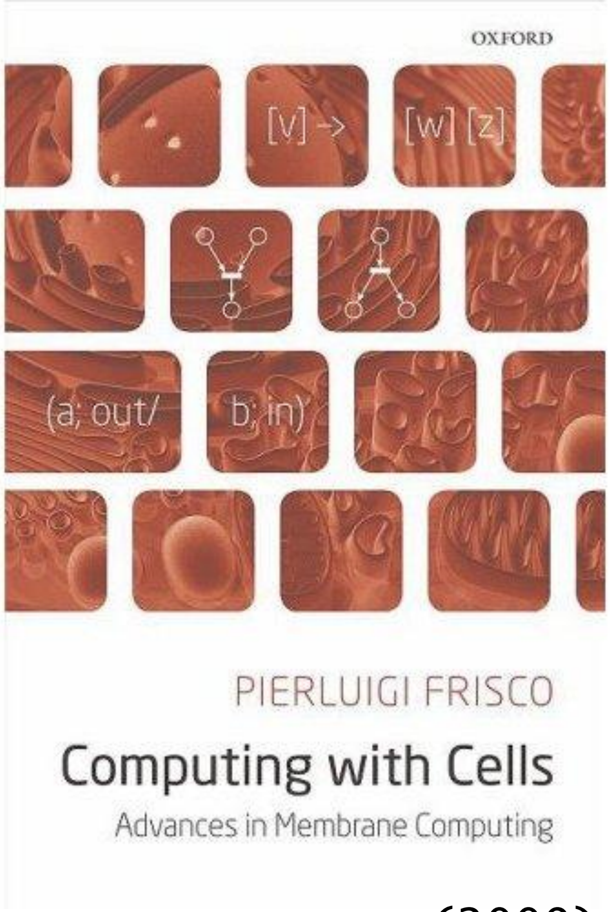
Natural Computing Series

Springer 2002
ISBN 3-540-42601-4

recent books

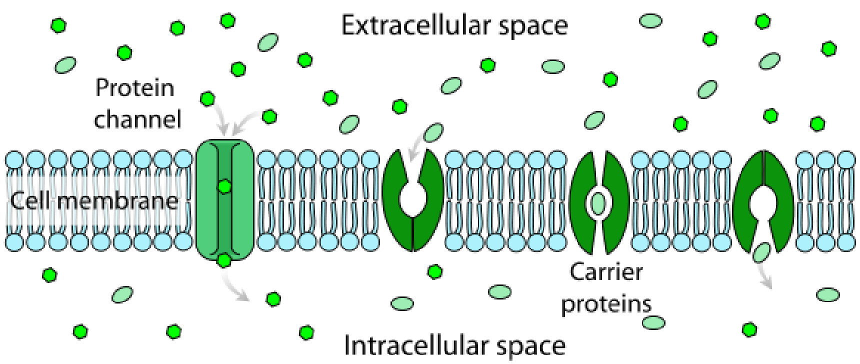


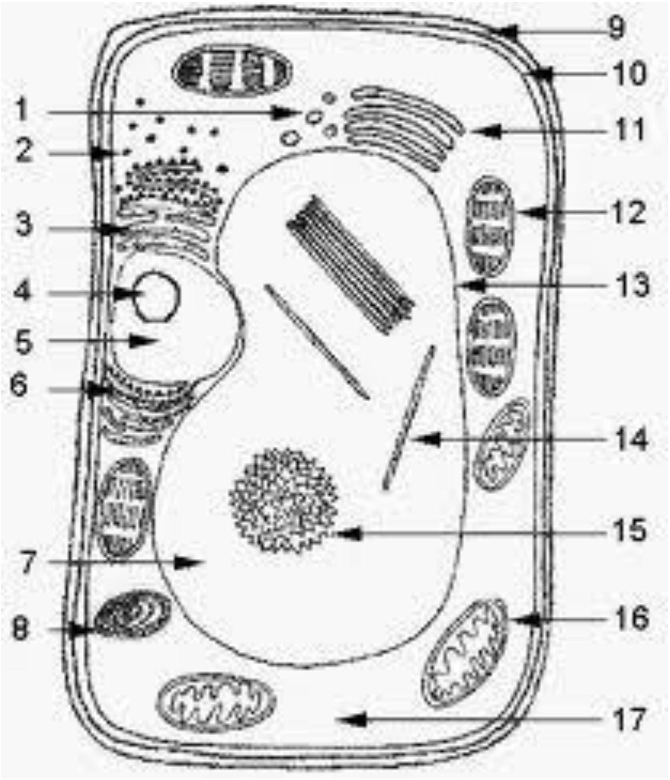
(2010)



(2009)

nature inspires





nested compartments
- information
membranes – communication

Places & Tokens

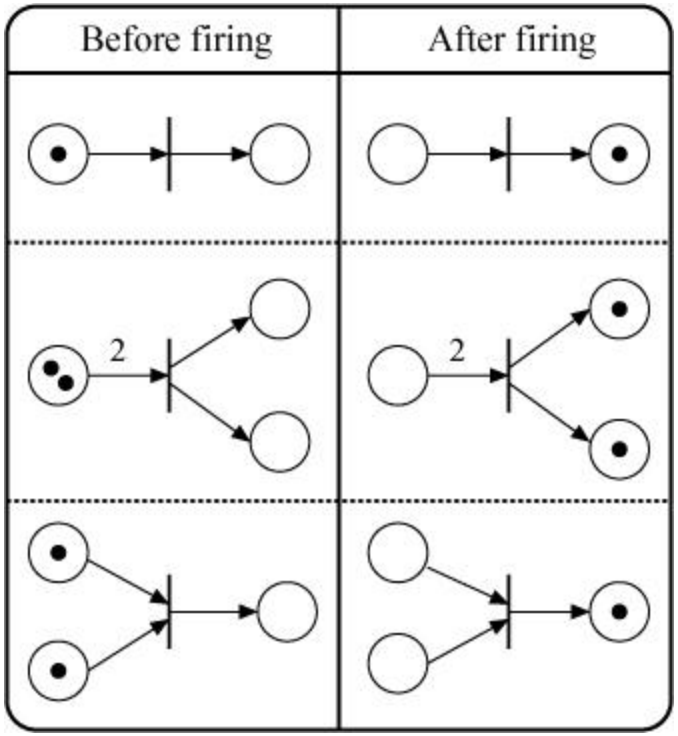
before that we had Petri nets

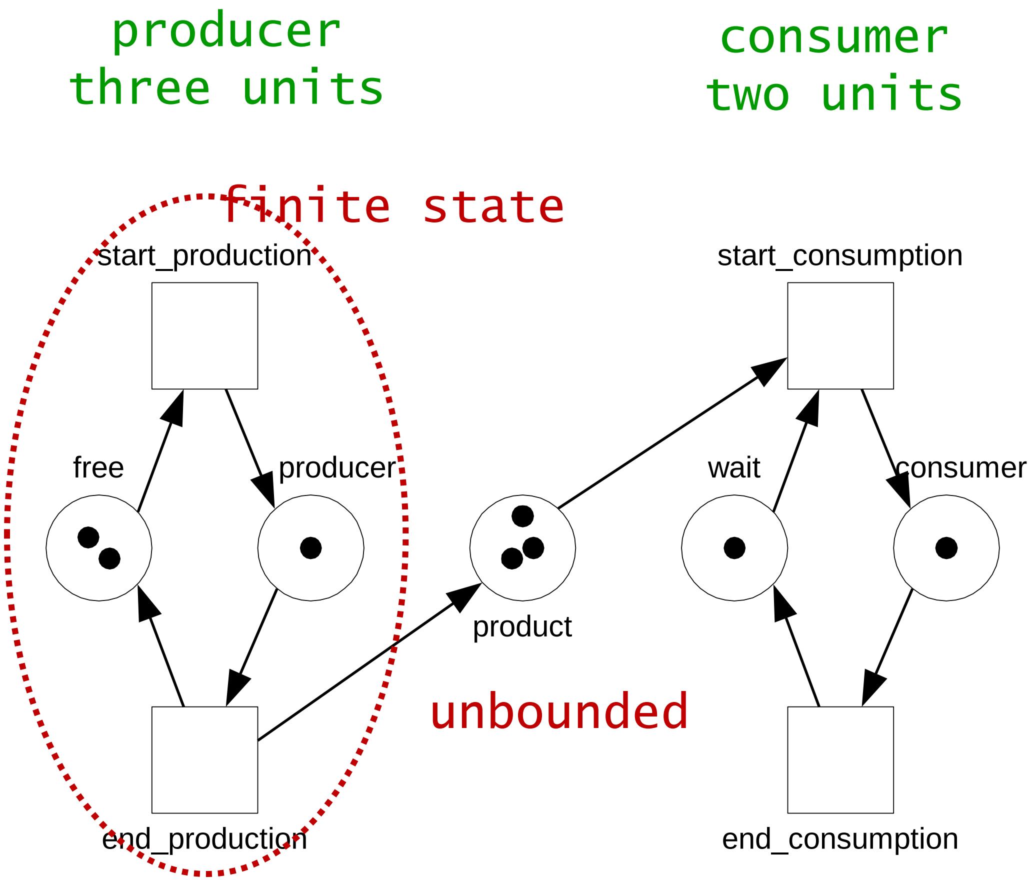
petri nets

circles → places + tokens → resources

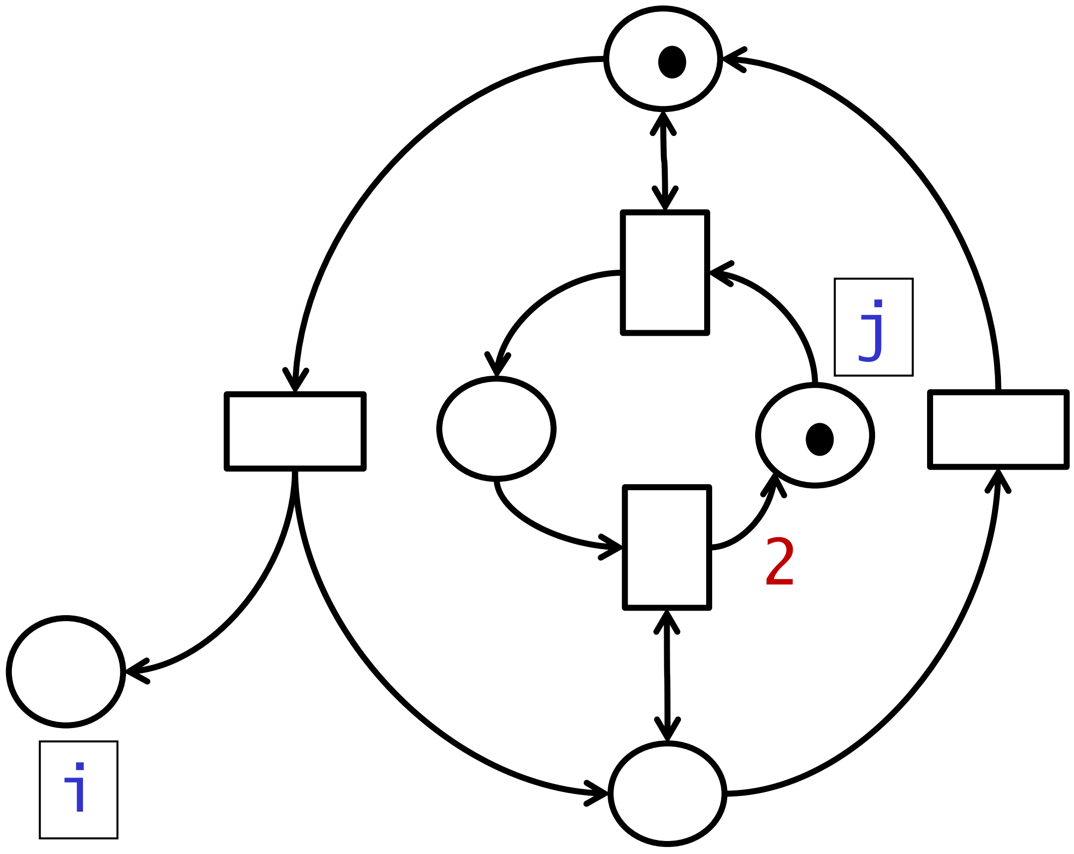
boxes → transitions → actions

circles	places + tokens	resources
boxes	transitions	actions



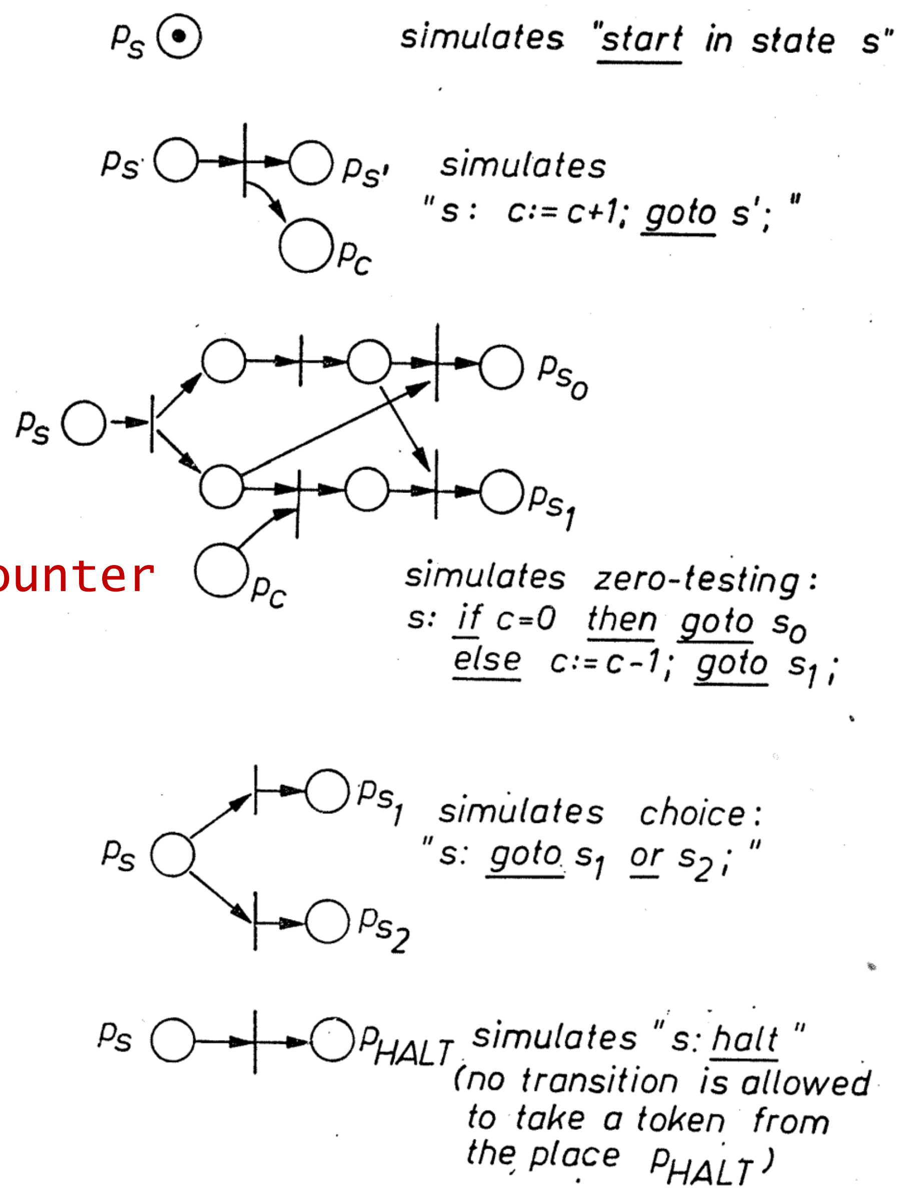


computing with petri net

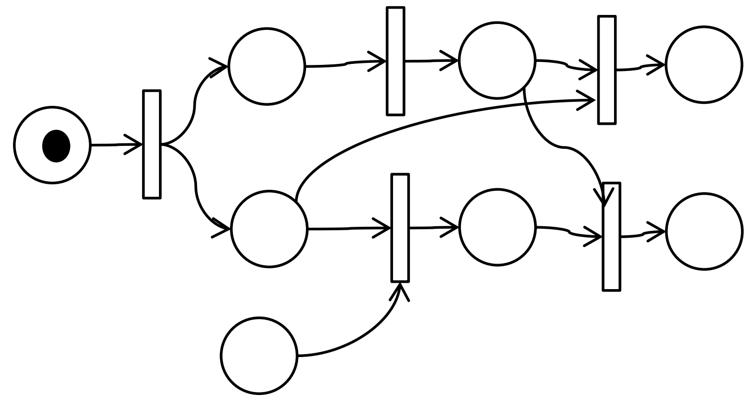


$\{ (i,j) \mid j \leq 2^{i} \}$

maximal strategy



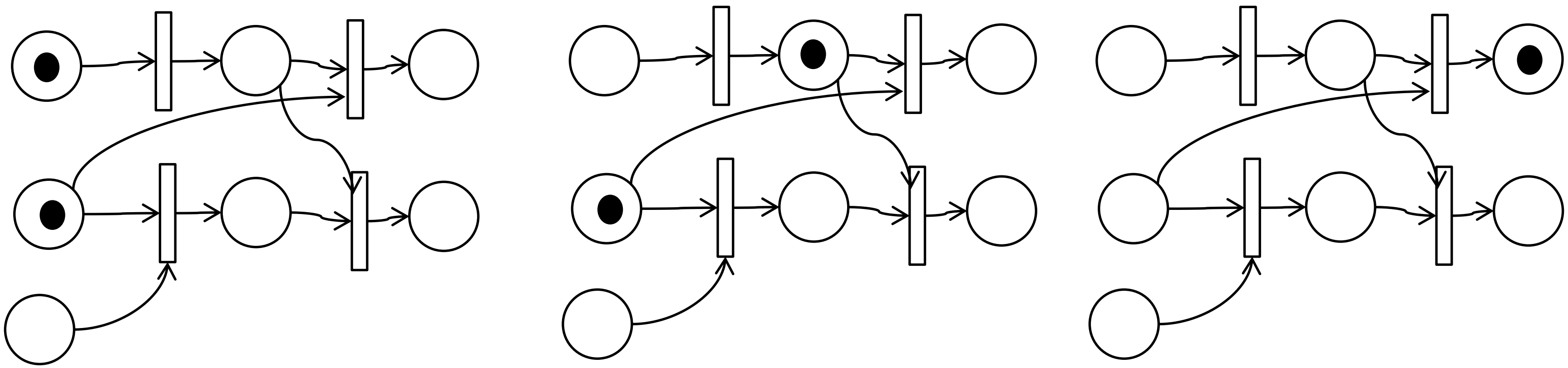
counter



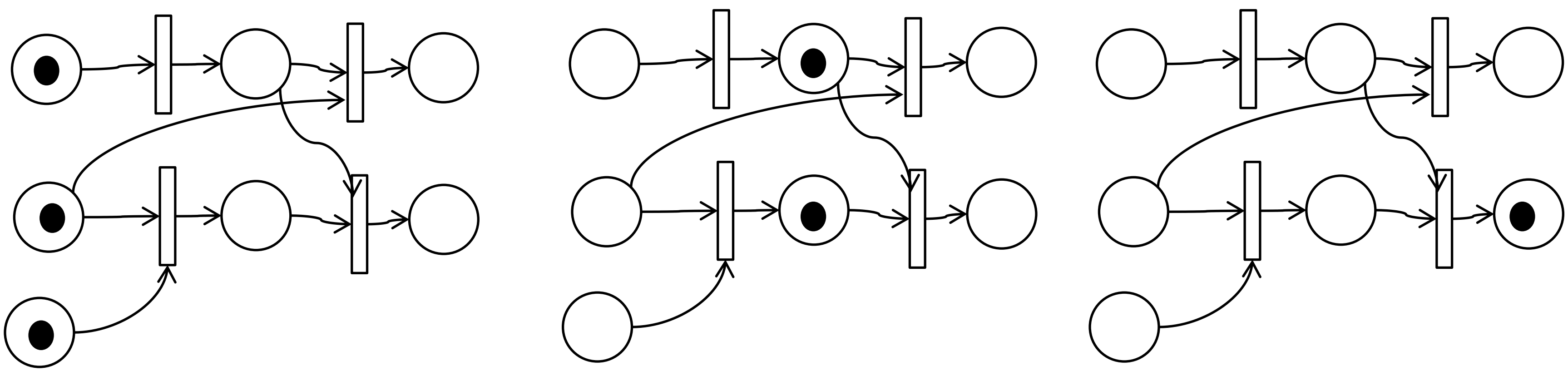
counter

Burkhard – Ordered firing in Petri Nets, EIK 1981

forced steps



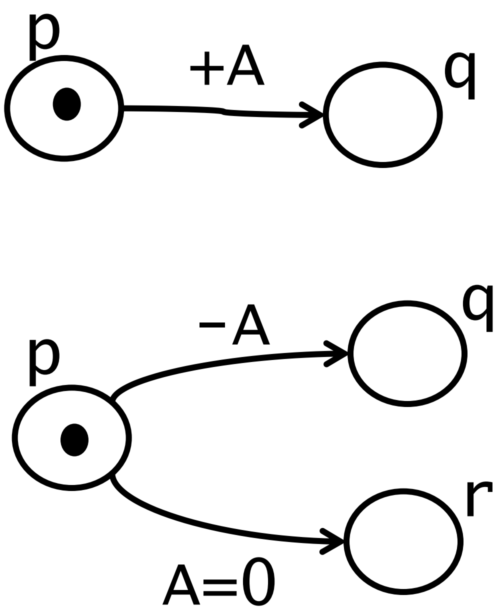
counter empty



counter
at least one token

where is max par important?

about counters



counter automaton =

register machine

Minski, Fischer

FRACTRAN

states + instructions

+A , -A , A=0 zero test

petri net =

'partially blind' counter automaton

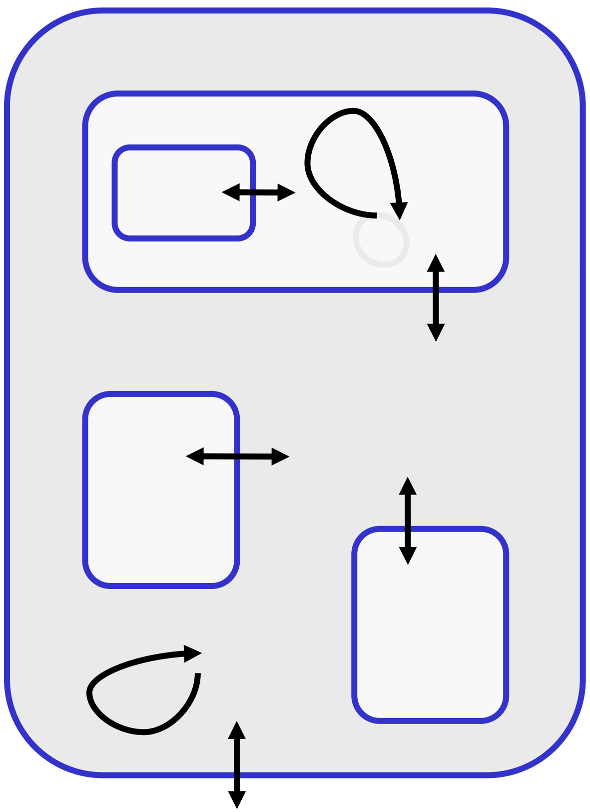
(*no zero test*)

Greibach, Petri

Membranes & Objects

Păun 2000 JCSS

Membrane Computing



nested membranes

objects

- strings
- unstructured

rules

- communication

  $ac \to ba_{out}c_{in}$

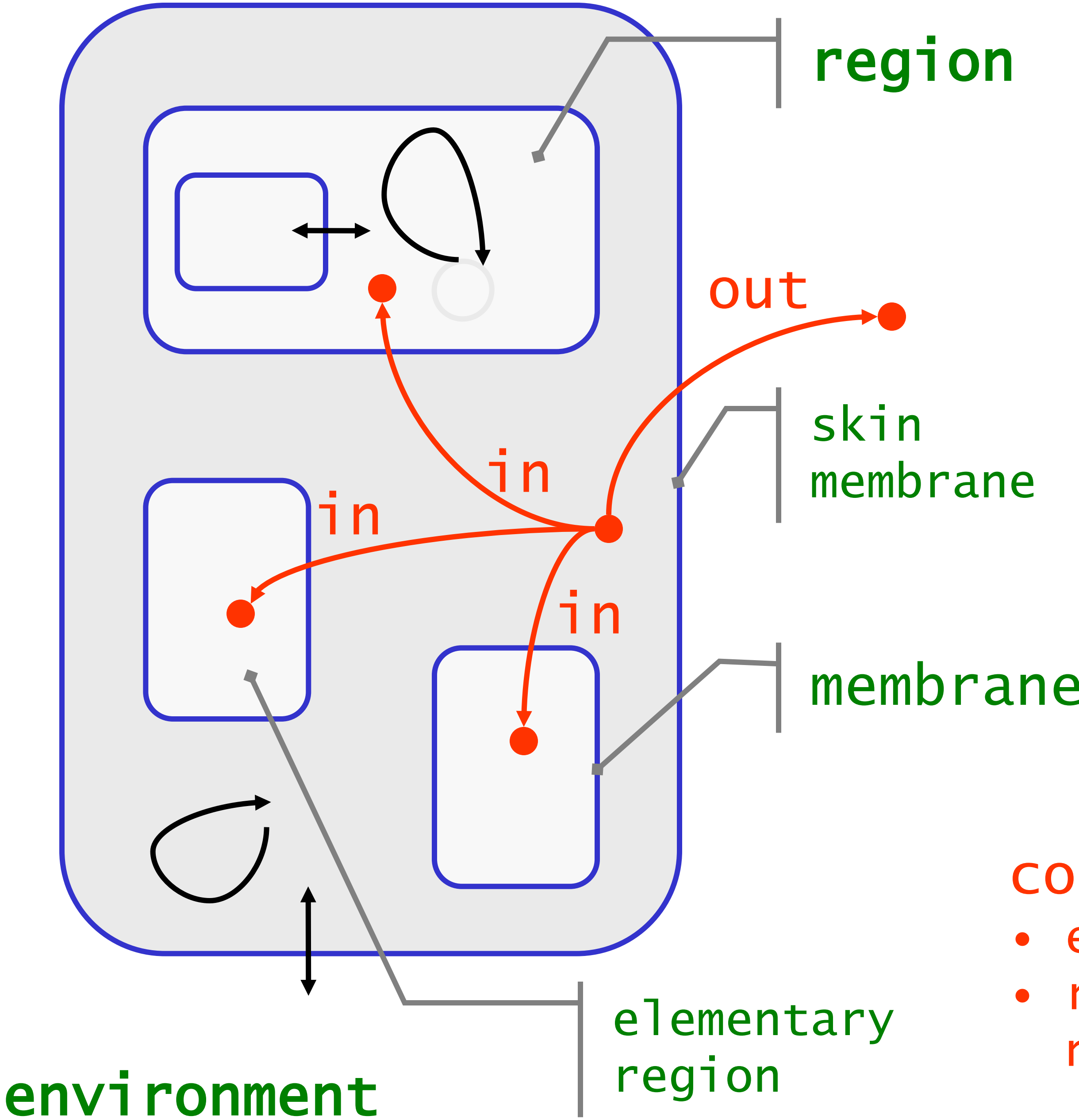
- evolution

  rewriting

  $(A \to aAb)_{out}$

  splicing

Structure



communication
- explicit addressing
- relative in/out
  non-deterministic

what is generated?

where is the input/output?

objects inside membrane:
(vectors of) natural numbers

- acceptance of input (by halting)

- generating output ‘enumerable’
  needs nondeterminism

Carriers are usually saturable monomeric proteins, which bind the transported solutions with high specificity, and move them at low flow rates. Channels are usually oligomeric complexes (with helical transmembrane domains or barrel-shaped structures), which show less stereospecificity than carriers but larger transport rates.

(p.38) Cell Biology for Membrane Computing

Carriers are usually saturable monomeric proteins, which bind the transported solutions with high specificity, and move them at low flow rates. Channels are usually oligomeric complexes (with helical transmembrane domains or barrel-shaped structures), which show less stereospecificity than carriers but larger transport rates.

(p.38) Cell Biology for Membrane Computing

Carriers & Objects

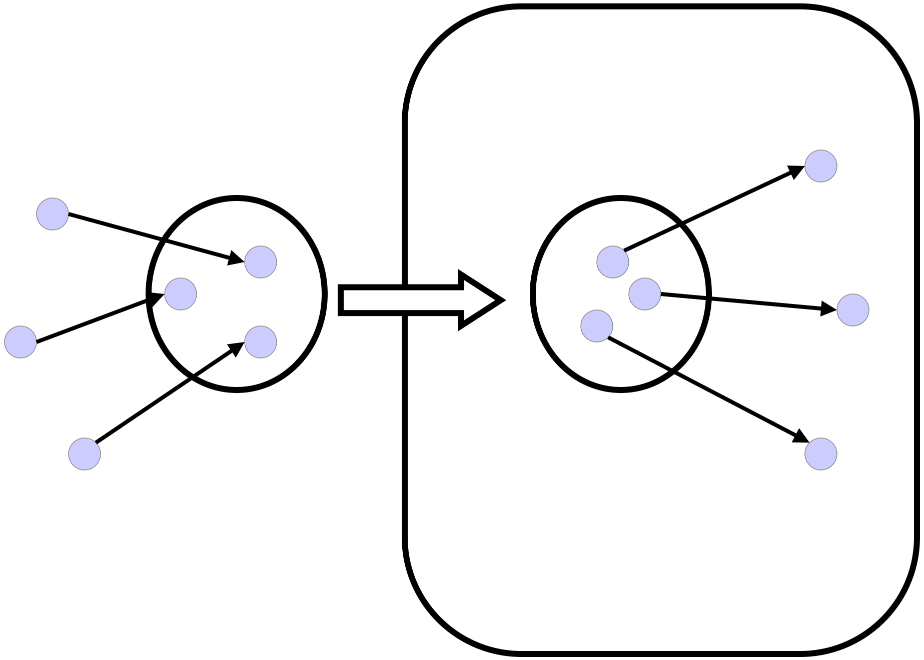
pure communication
maximal parallellism

Carriers and Counters (DLT2002)

P systems with Carriers vs. (Blind) Counter Automata

P systems with carriers

Martín-Vide Păun Rozenberg



Contents

- objects
  - multiset symbols
  - infinite supply in environment
- carriers
  - finite number

Rules

$va_1 \ldots a_k \rightarrow v[a_1 \ldots a_k]$ attaching

$v[a_1 \ldots a_k] \rightarrow in$ carry in

$v[a_1 \ldots a_k] \rightarrow out$ carry out

$v[a_1 \ldots a_k] \rightarrow va_1 \ldots a_k$ detaching

$va_1 \ldots a_k \leftrightarrow v[a_1 \ldots a_k]$

$v[a_1 \ldots a_k] \rightarrow in/out$

Computations

evolving multisets

infinite supply environment

fixed carriers

maximal parallellism

halting by ‘blocking’

counting objects

‘output’ membrane

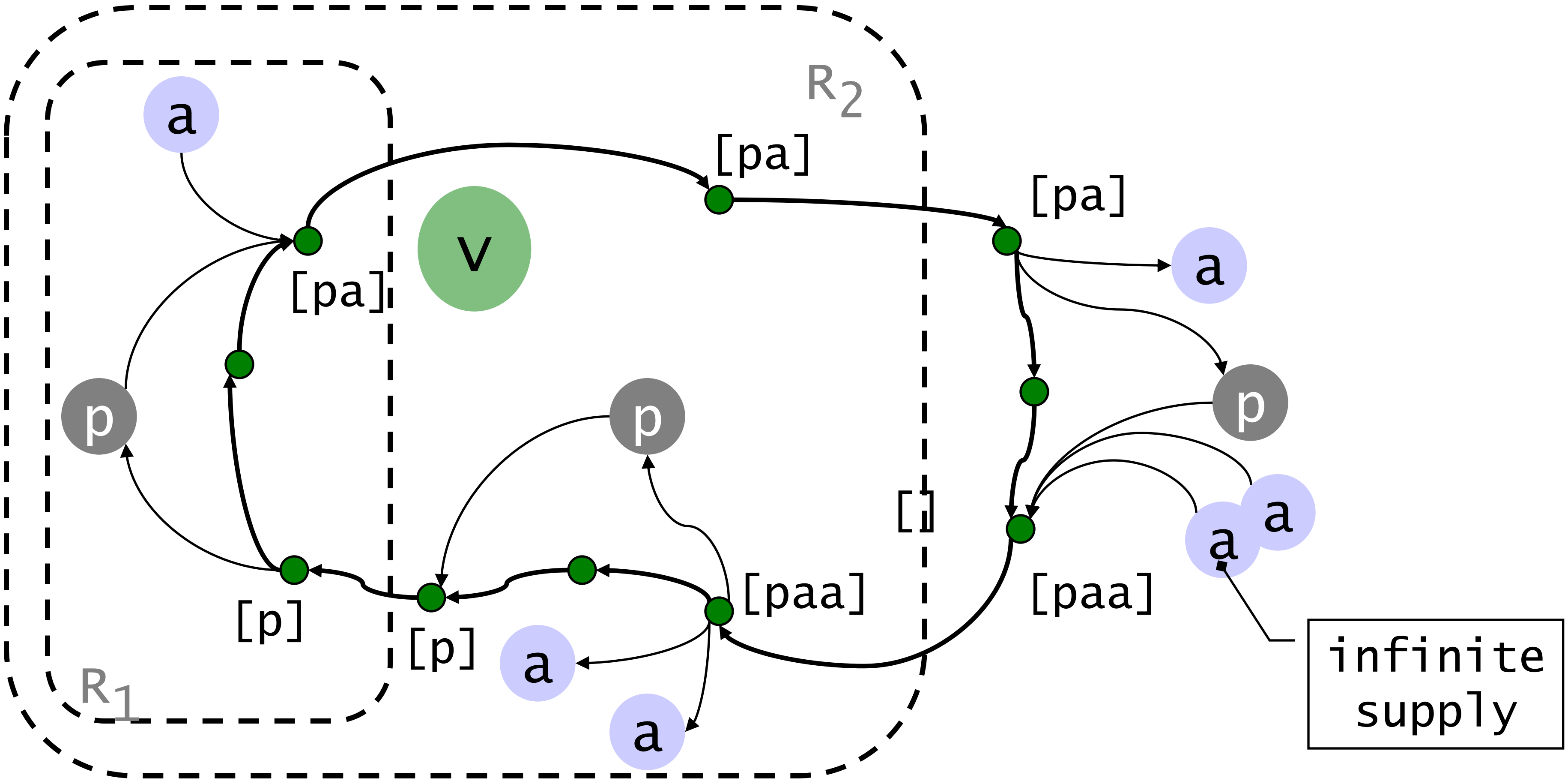
$\mathbb{N}^k CP_m(c,p)$

- membranes
- carriers
- passengers

(here k=1)

P systems with carriers

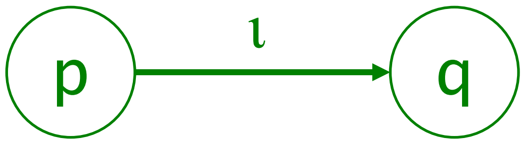
example



E	R_2	R_1
	$vp \to v[p] \to in$	$v[p] \to vp$
$v[pa] \to vpa$	$v[pa] \to out$	$vpa \to v[pa] \to out$
$vpaa \to v[paa] \to in$	$v[paa] \to vpaa$	

counter automata

Minsky, Fischer



$(p \rightarrow q, \iota)$

ε	nil
$+A$	add one
$-A$	substract one
$A=0$	zero test

several counters
acceptance by final state
& empty counters

$\mathbb{N}$RE

output counter

Recursively Enumerable sets (of numbers)

blind counter automata

Greibach

no zero test,
except final test for acceptance
Petri nets

$\mathbb{N}$BC

DLT'02 paper

MaVi-Pău-Roz '02

$$\boxed{\mathbb{N}RE = \mathbb{N}CP_2(3,3)}$$

$$\mathbb{N}CP_m(c,p)$$

- membranes
- carriers
- passengers

$$\mathbb{N}RE = NCP_1(2,3)$$
$$= NCP_1(*,2)$$

1. single membrane

$$\mathbb{N}BC = NCP_*(1,*)$$
$$= NCP_1(1,3)$$

2. single carrier

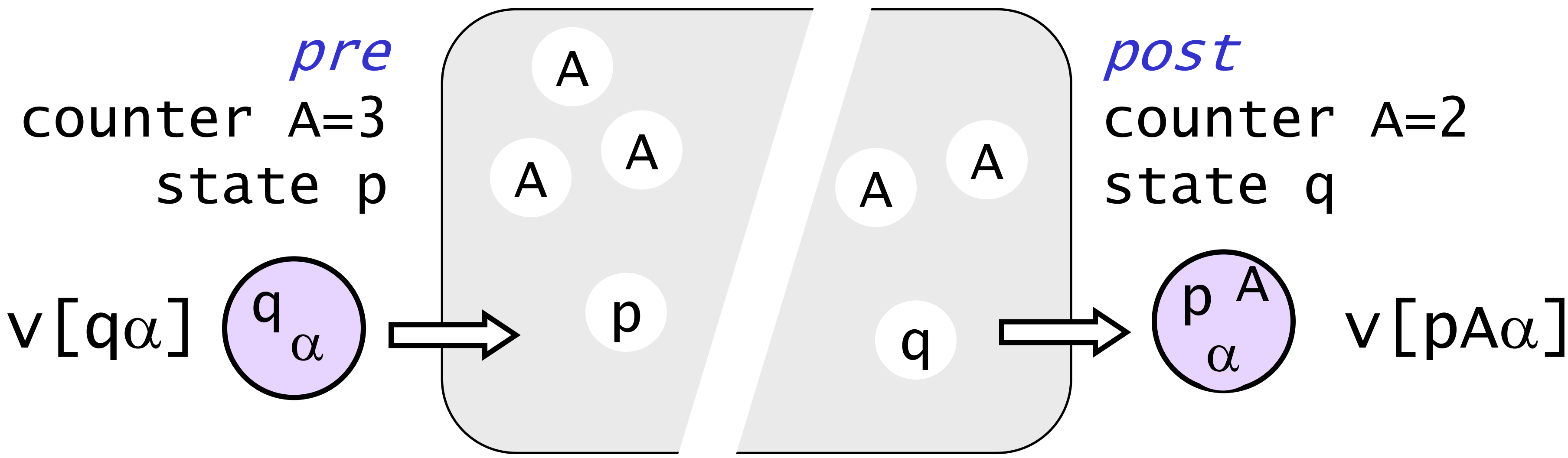
$$\mathbb{N}RE = NCP_*(*,1)$$

3. single passenger

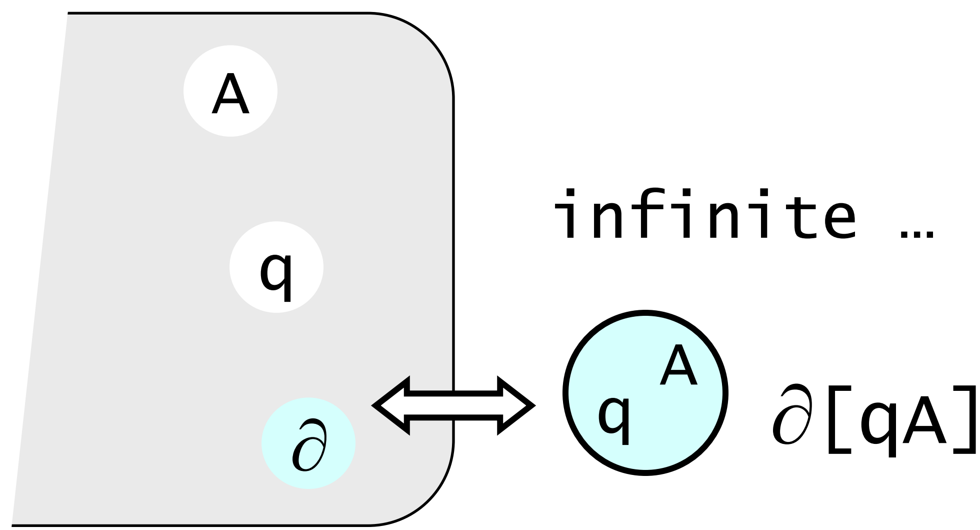
1. single membrane

$\mathbb{N}\mathbf{RE} \subseteq \mathbb{N}\mathbf{CP}_1(2,3)$

$\alpha:(p \rightarrow q, -A)$



$\alpha:(p \rightarrow q, A=0)$



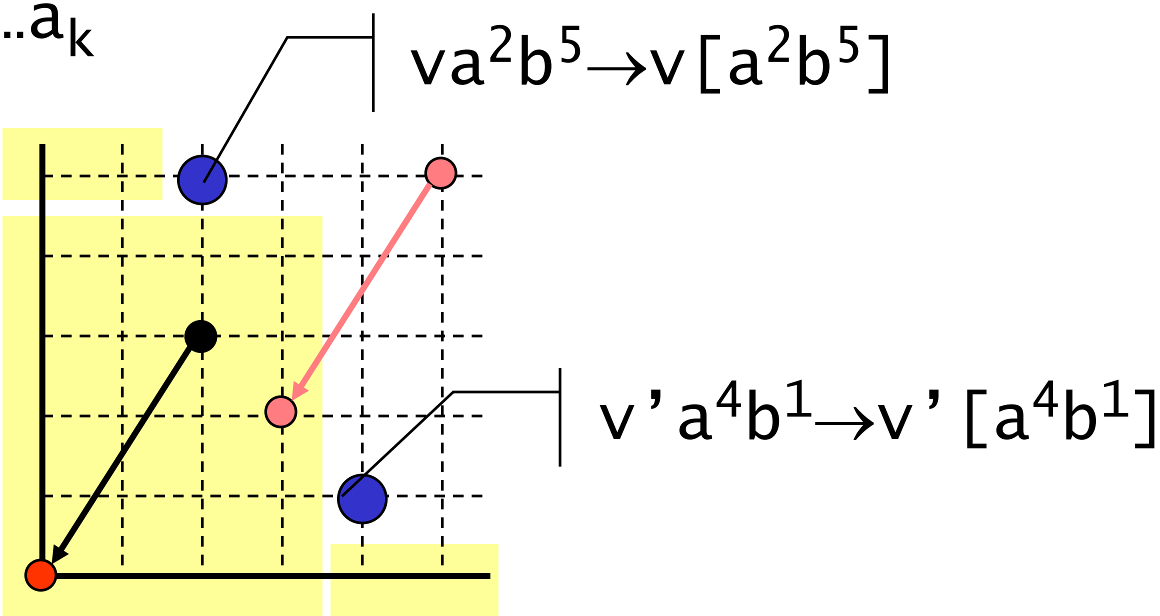
2. single carrier

P without parallellism
carriers + objects
P system halting:
no applicable rules

blind counter aut.
state + counters
final state
& empty counters

- $va_1 \ldots a_k \to v[a_1 \ldots a_k]$
- $v[a_1 \ldots a_k] \to$ in/out
- $v[a_1 \ldots a_k] \to va_1 \ldots a_k$

guess vector
& test by zero
acceptance



2. single carrier

$\boxed{\mathbb{N}\mathrm{BC} = \mathbb{N}\mathrm{CP}_1(1,3)}$

$$\mathbb{N}\mathrm{BC} \subseteq \mathbb{N}\mathrm{CP}_1(1,3)$$

$$\mathbb{N}\mathrm{RE} \subseteq \mathbb{N}\mathrm{CP}_1(2,3)$$

forget about 'zero test' ∂

$$\mathbb{N}\mathrm{BC} \supseteq \mathbb{N}\mathrm{CP}_*(1,*)$$

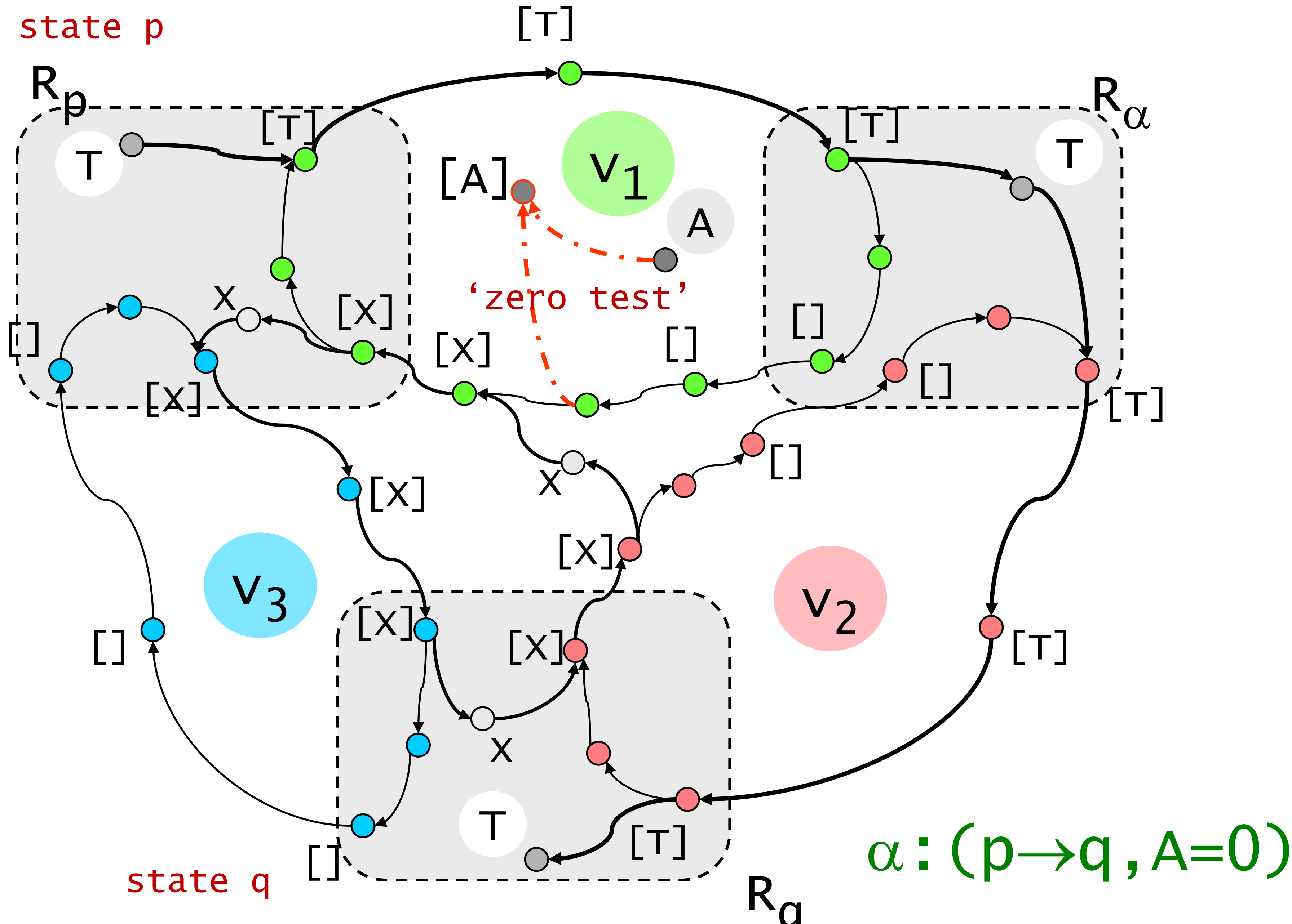
no parallellism !

$\mathbb{N}$BC single object semilinear $\rightarrow$ regular

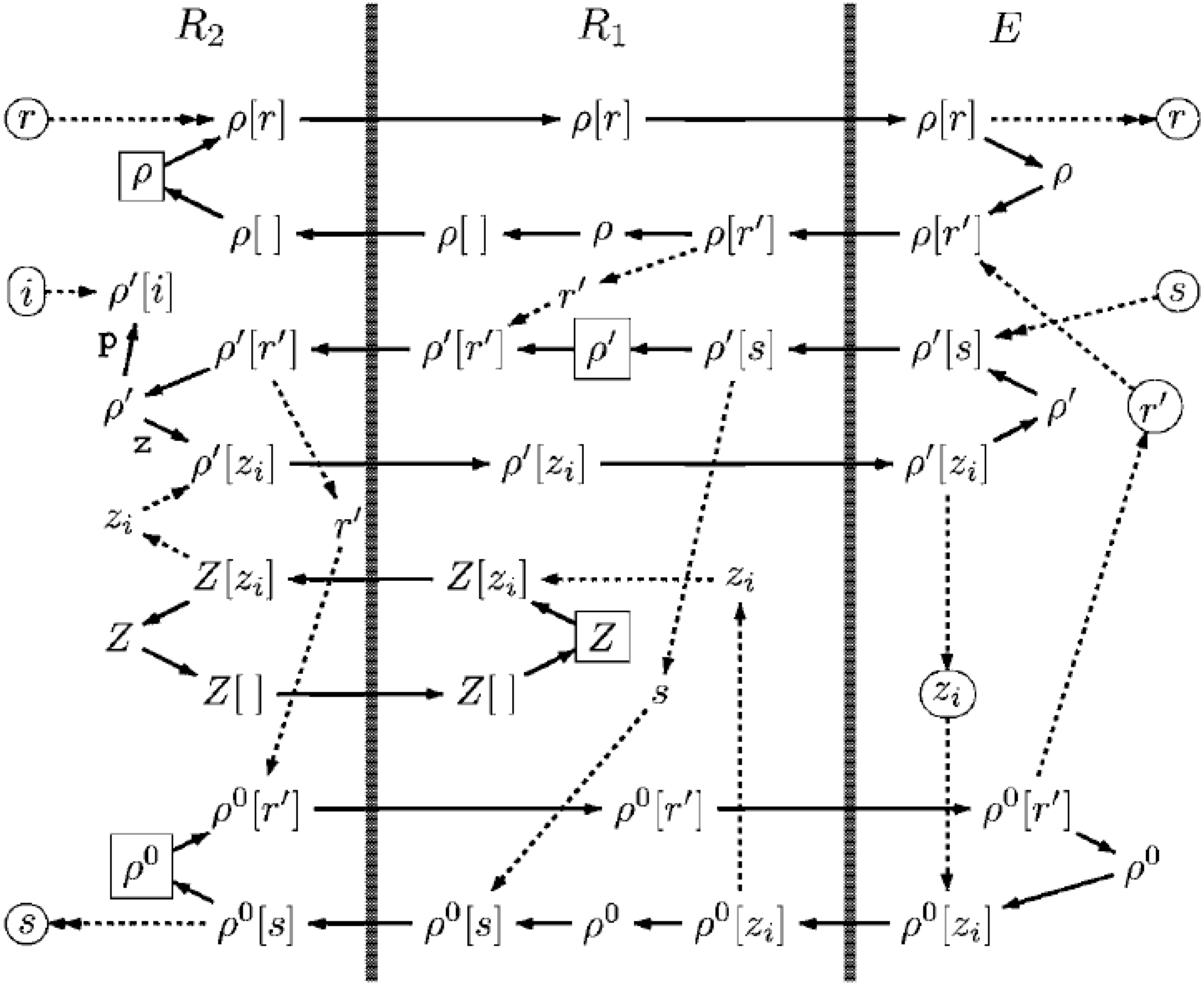
also with more objects $\{ (i,j) \mid j \leq 2^i \}$

3. single passenger

$\mathbb{N}\mathrm{RE} \subseteq \mathbb{N}\mathrm{CP}_*(*,1)$

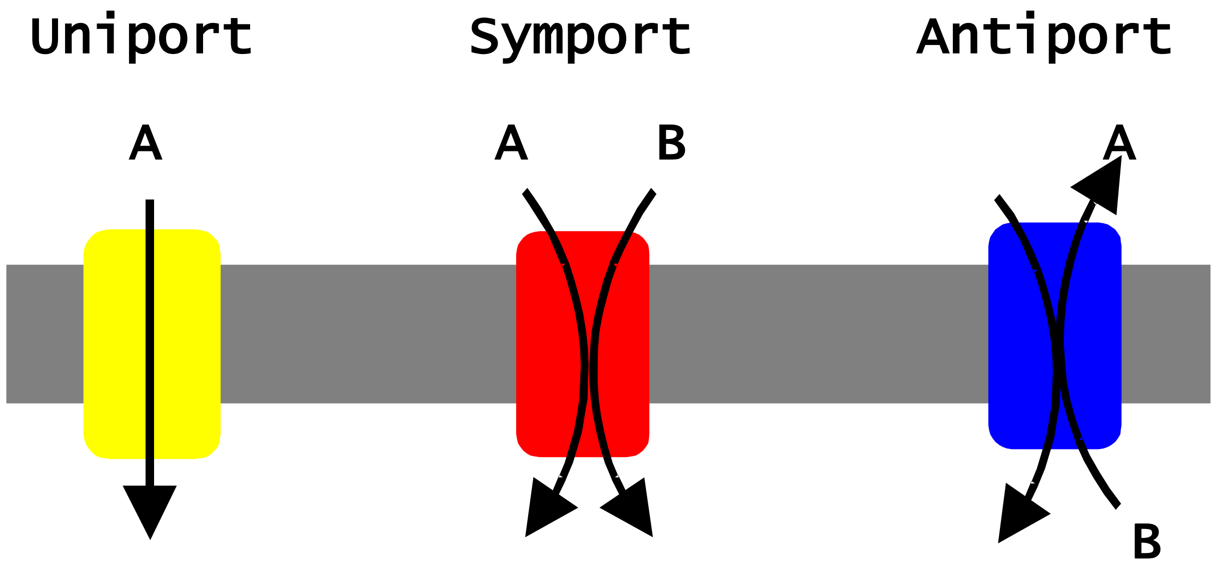


two membranes is ok



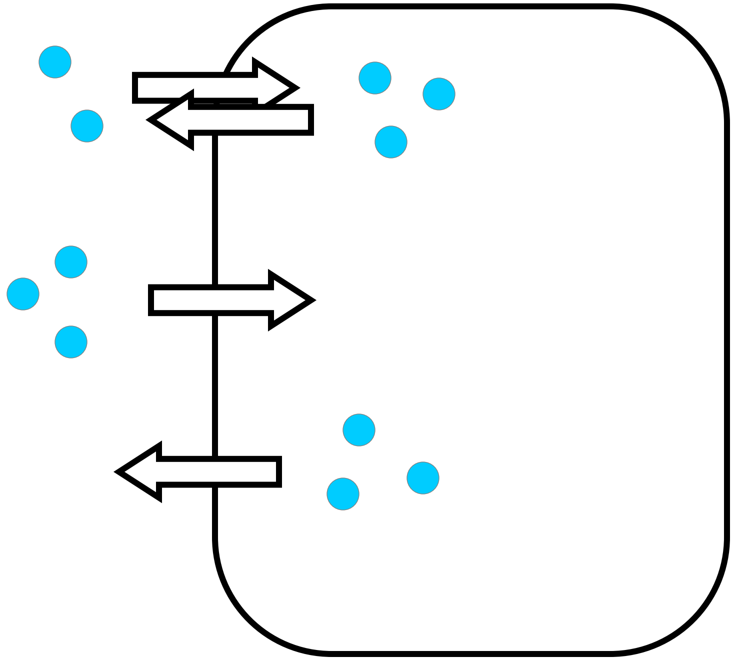
$\mathbb{N}RE = NCP_2(*, 1)$

Symport & Antiport


Uniport
Symport
Antiport
A
A
B
A
B

P systems with symport/antiport

Păun & Păun



Contents

- objects
  - multiset symbols
  - infinite supply in environment

Rules

$(a_1 \ldots a_k, in; b_1 \ldots b_\ell, out)$ antiport

$(a_1 \ldots a_k, in)$ symport

$(a_1 \ldots a_k, out)$

WMC paper (& Frisco)

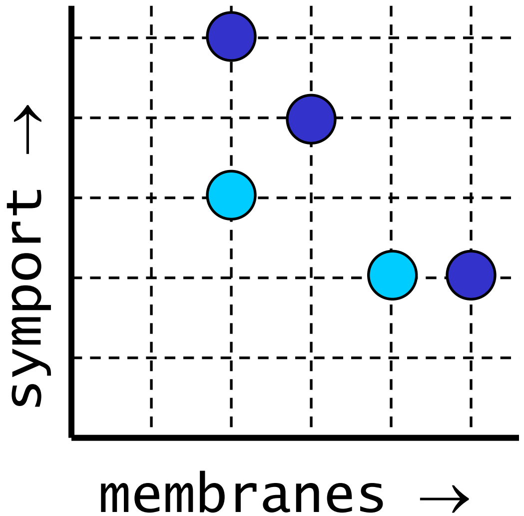
Pău-Pău '02

$\boxed{\mathbb{N}RE = \mathbb{N}PP_2(2,2)}$

$\mathbb{N}PP_m(s,a)$

- membranes
- symport
- antiport

$\mathbb{N}RE = \mathbb{N}PP_1(1,2)$



1. single membrane

2. symport only

programming trick

$\mathbb{N}RE = \mathbb{N}PP_1(1,2)$

good vs. bad ?
infinite
blocking

conflicting counters A & A’

(p→q, A=0)

(p→r, +A’)
(r→r’, ϵ)
(r’→q, -A’)

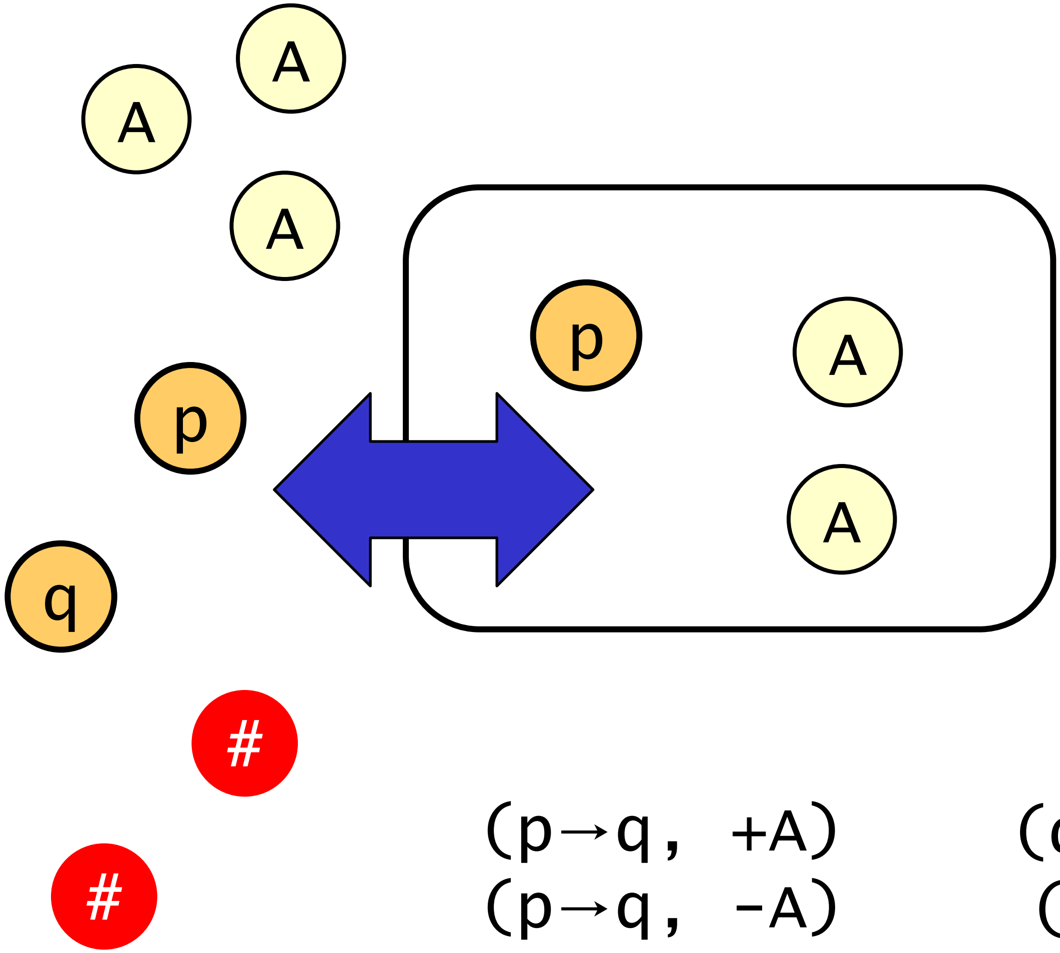
(#,in;AA’,out)
(#,in;#,out)

counter aut / new antiport

max parallellism: forced move

single membrane will do

sadly enough



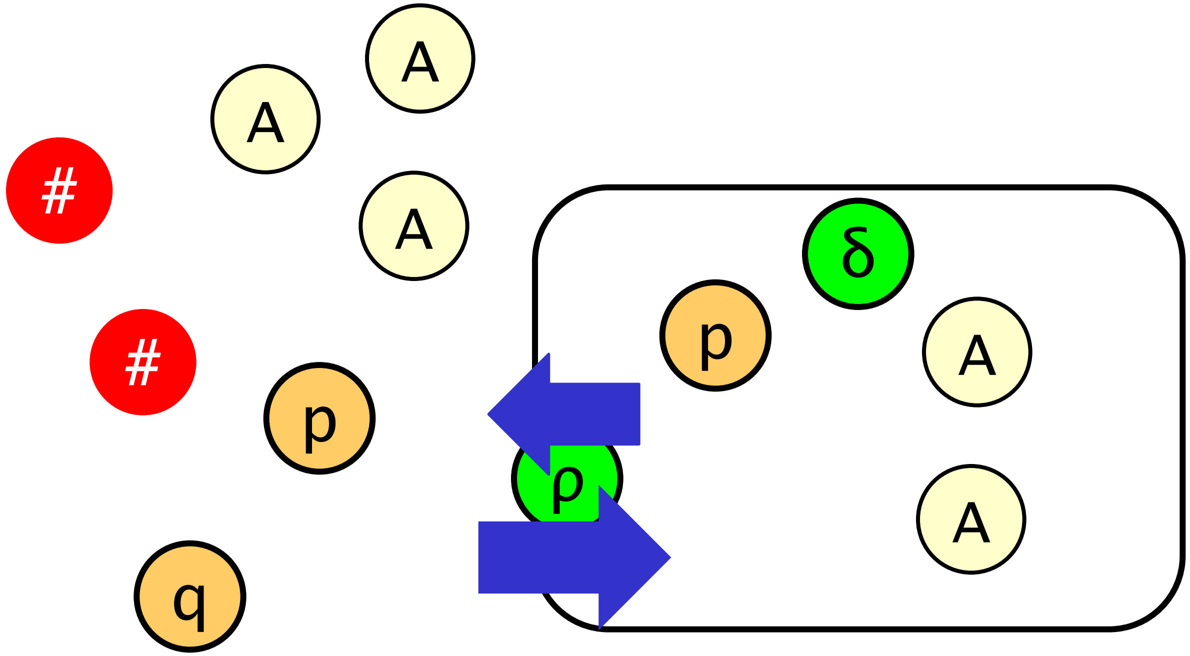
$(p \to q, +A)$ $(qA, in; p, out)$

$(p \to q, -A)$ $(q, in; pA, out)$

counter aut antiport

antiport to symport

add a ‘carrier’

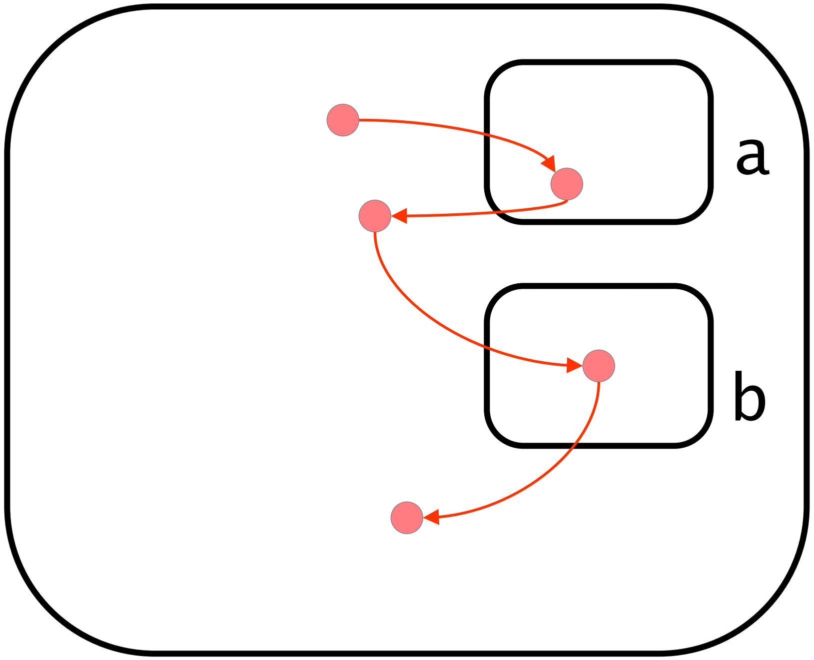


technical: halting state (extra symbol)

$(p \to q,\ +A)$	$(qA,in;p,out)$	$(\rho qA,in)(\rho p,out)$
$(p \to q,\ -A)$	$(q,in;pA,out)$	$(\rho q,in)(\rho pA,out)$
counter aut	antiport	symport

symport/antiport : following the traces

Ionescu, MartínVide, Păun & Păun



Contents

- objects
  multiset symbols
  infinite supply
  in environment
- traveller

Rules

$(a_1 \ldots a_k, in; \; b_1 \ldots b_\ell, out)$ antiport

$(a_1 \ldots a_k, in)$ symport

$(a_1 \ldots a_k, out)$

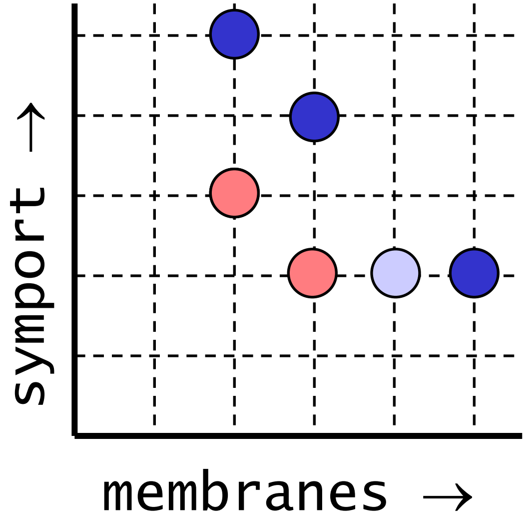
WMC paper

Ion-MaVi-Pău-Pău '02

$$\boxed{1\cdot RE = 1\cdot LP_2(2,2)}$$

$$\ell\cdot RE = \ell\cdot LP_{\ell+1}(0,2)$$

$$\ell\cdot RE = \ell\cdot LP_{\ell+1}(3,0)$$



$\ell\cdot LP_m(s,a)$

- letters
- membranes
- symport
- antiport

1. two+ letters

2. single letter
   symport only

conclusion … after WMC'02 …

carrier P systems ↔ counter automata
maximal parallellism ↔ zero test

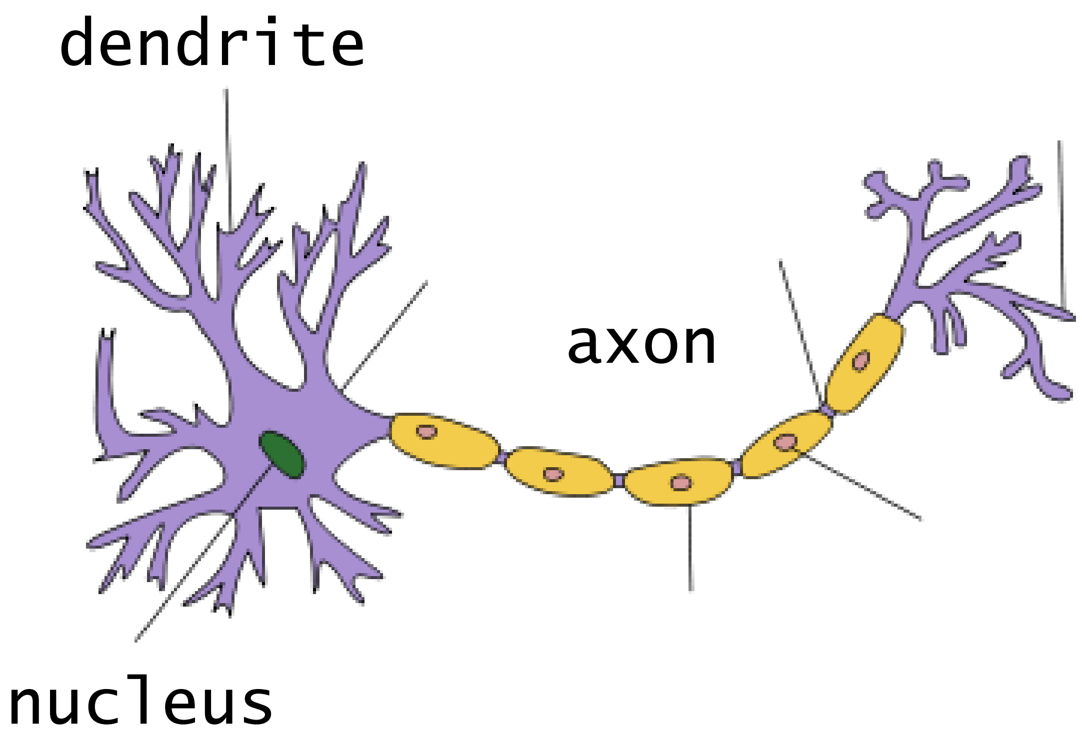
Petri nets!

single membrane	RE
single carrier	BC
single passenger	RE

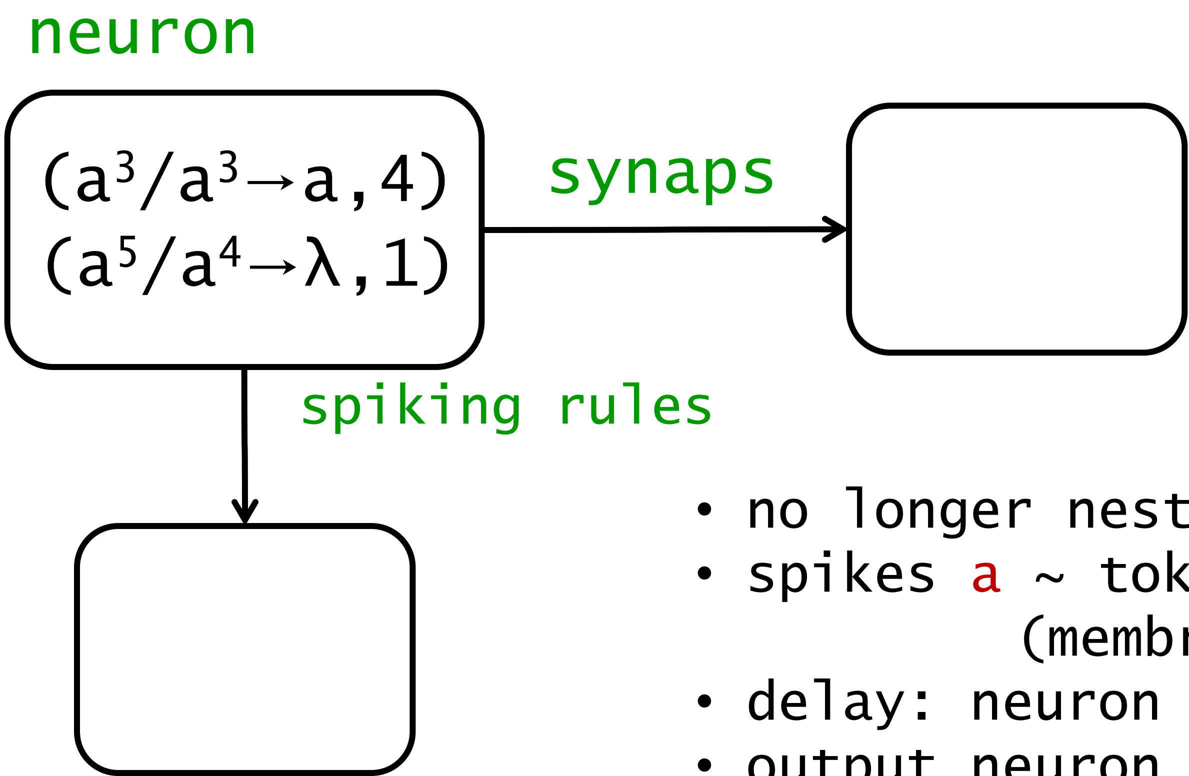
P systems with unstructured objects

- catalists & communicative [Sosík]
- P-automata [Csuhaj-Varjú & Vaszil]
- analysing systems [Freund & Oswald]
- with symport/antiport
- following traces

Neurons & Spiking


dendrite
axon
nucleus

neuron features

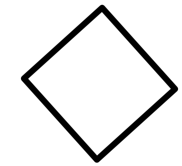


- no longer nested structure
- spikes a ~ tokens
  (membrane potential)
- delay: neuron closed
- output neuron
- time / synchronisation

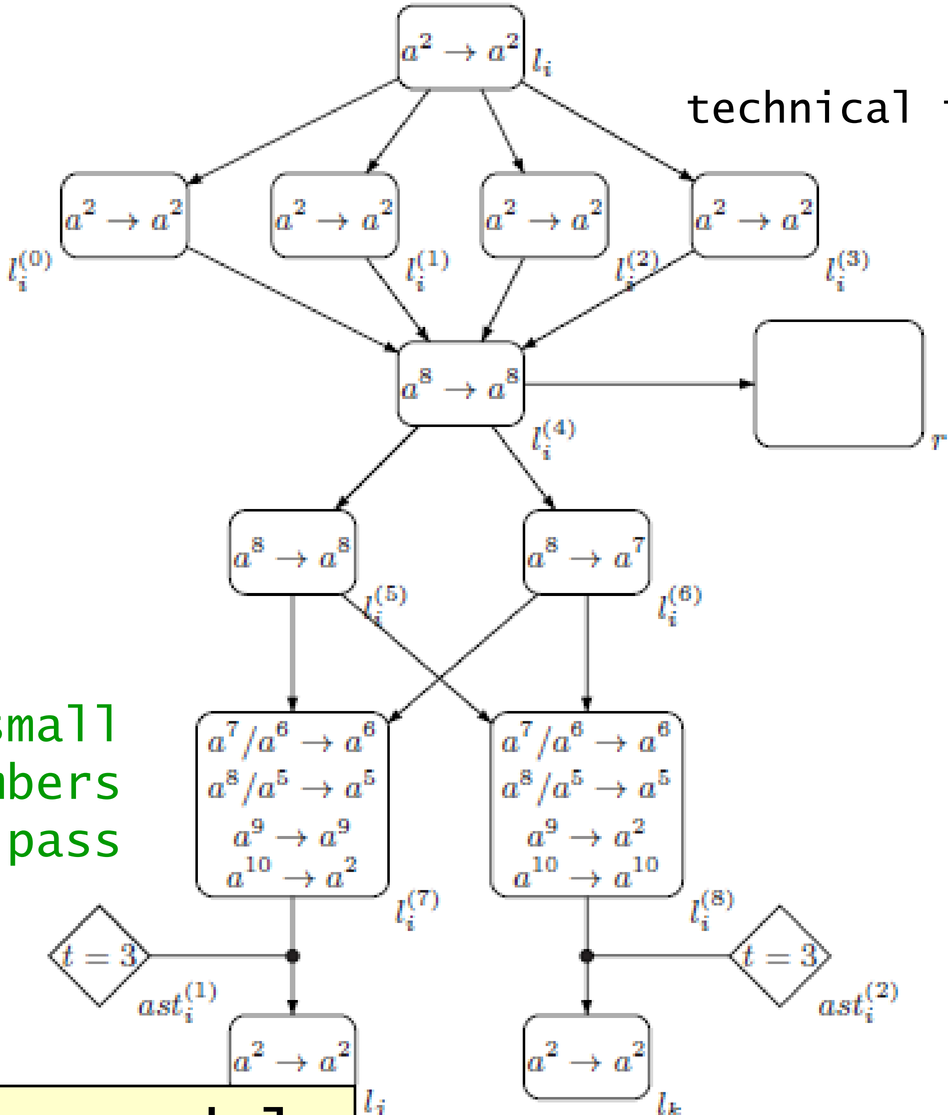
- astrocytes
- neuron division

astrocytes

block signals
at threshold



technical trick



counter

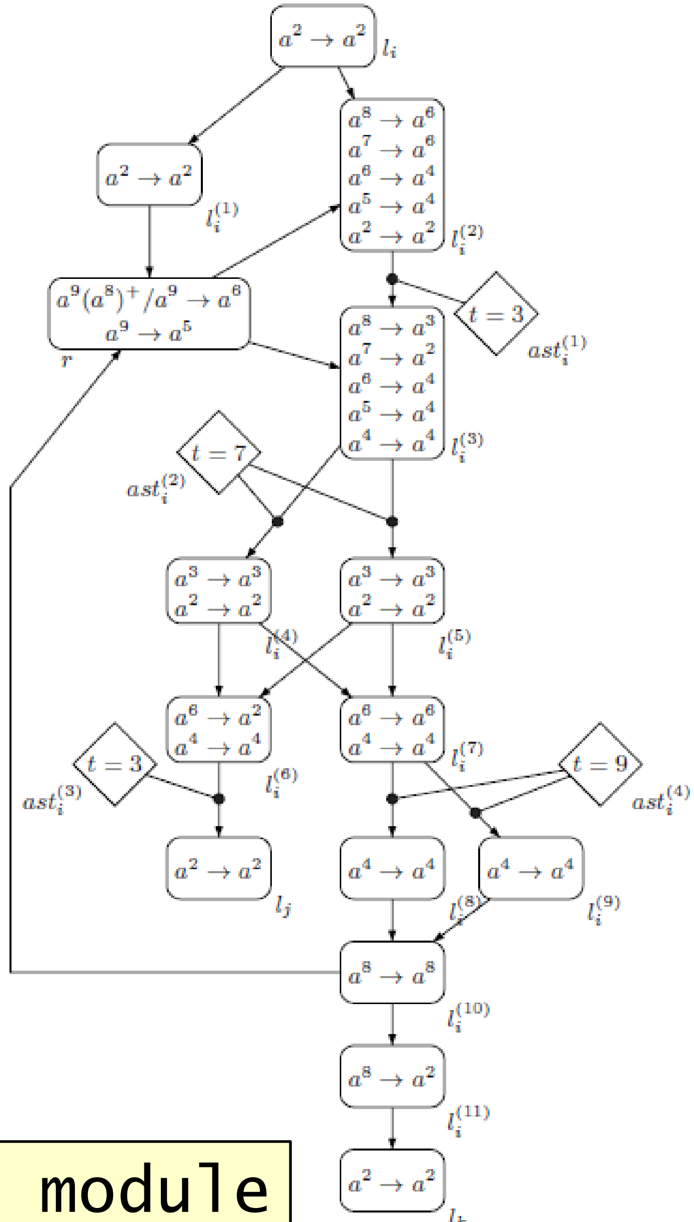
small
numbers
pass

ADD module

nondeterminism
through timing!

no maximality,
i.e.,asynchronous

module substract



SUB module

rules in ‘counter’-
membrane
never really empty

different instructions
same counter
[conflict!?]

not *forced* to clear up
may block

acceptance by
‘final state’

neuron division

‘solving’ NP complete problems

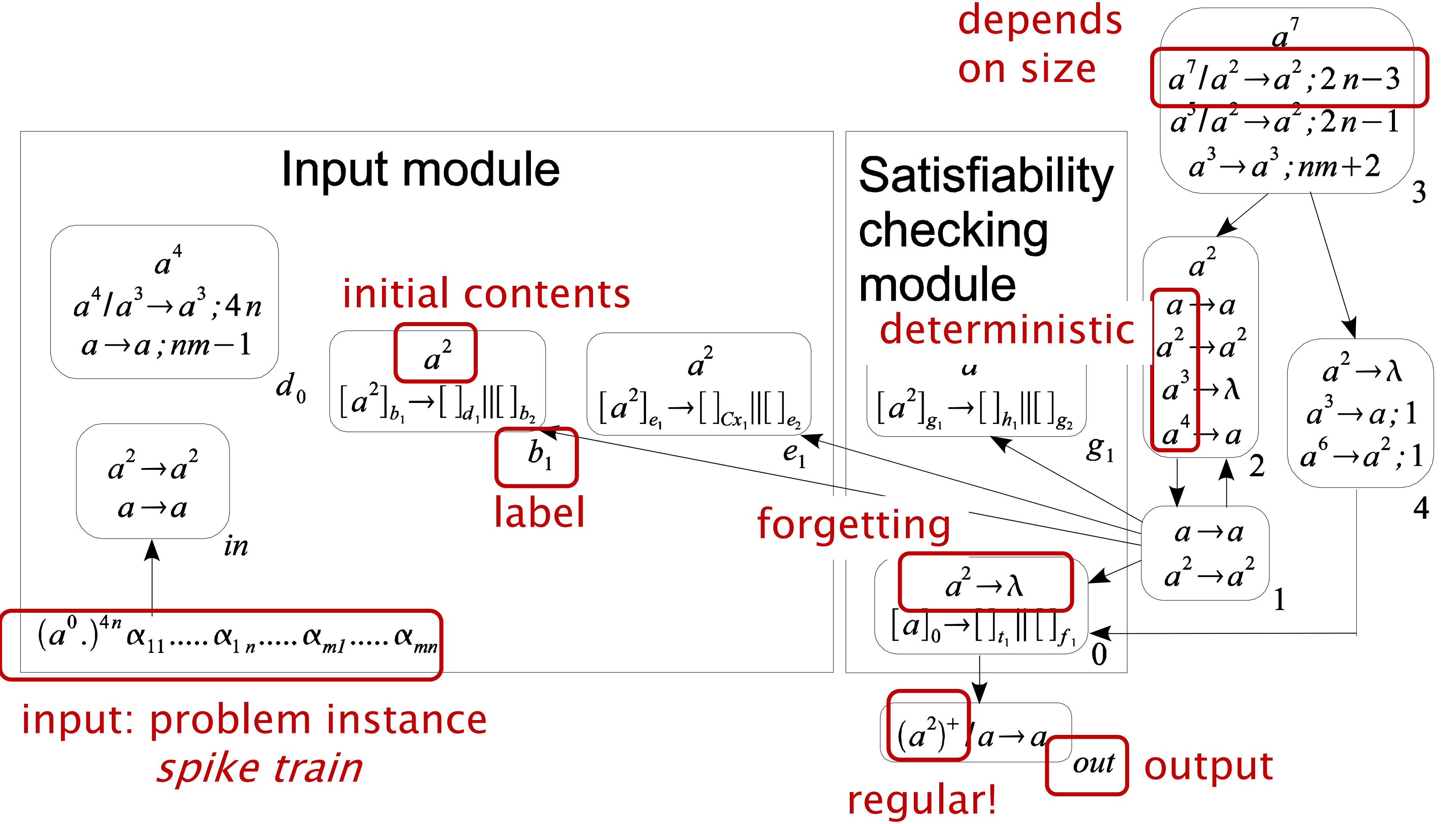
SAT satisfiability

input spike train = formula

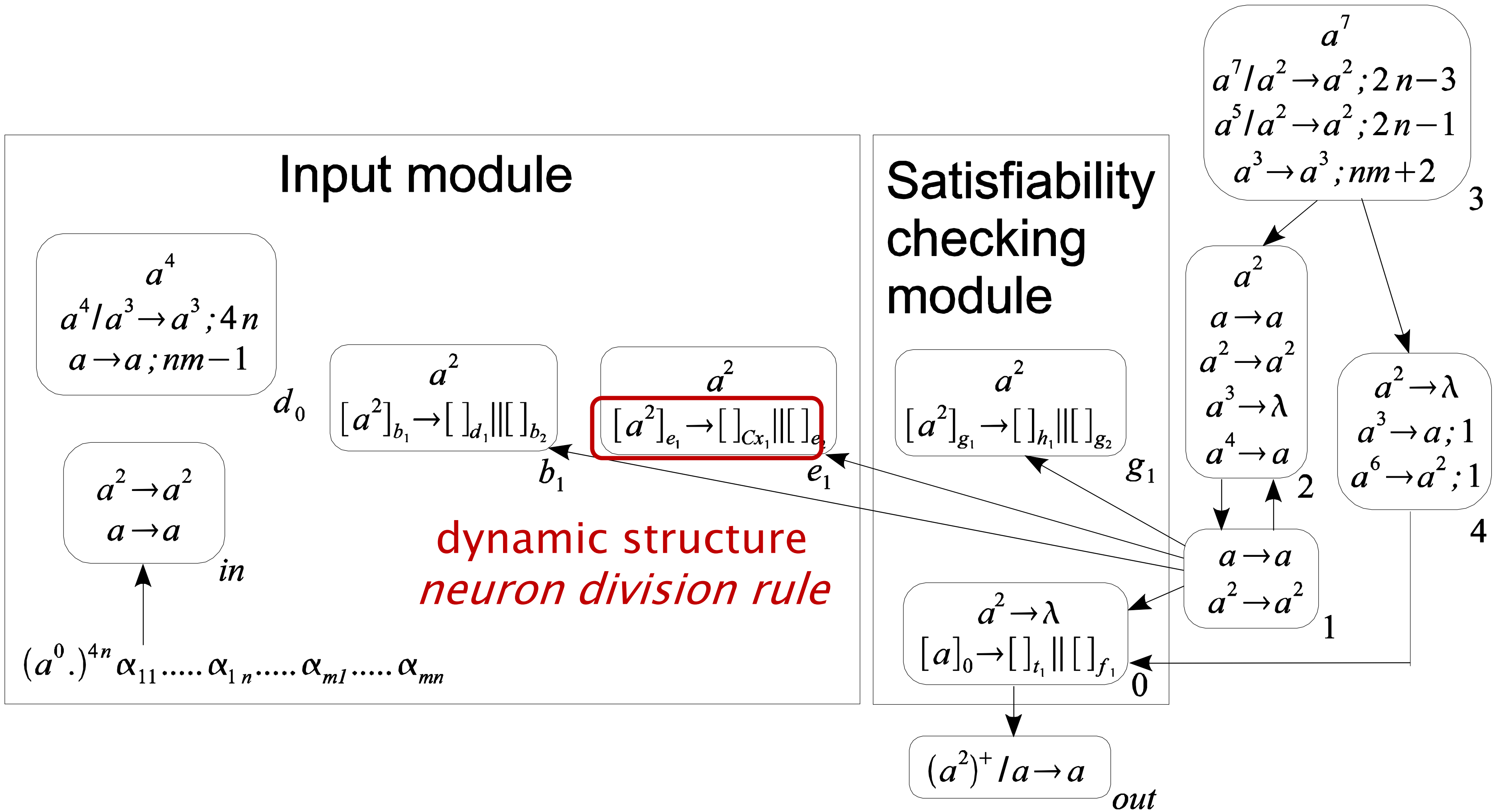
transformed into neuron structure

linear in time,
exponential in size

initial system



initial system



neuron division

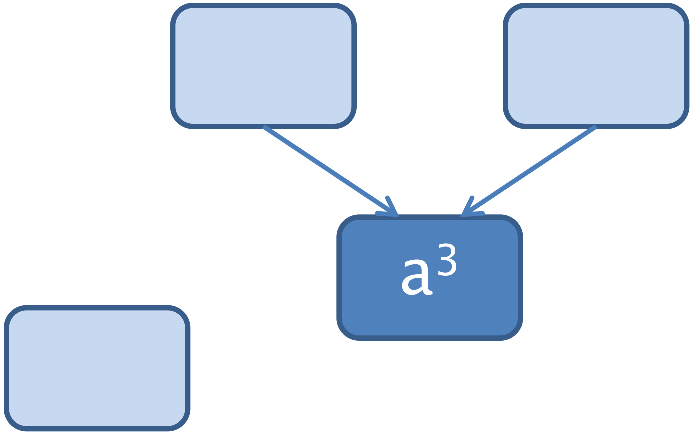
$$[\ E\ ]_i \rightarrow [\ ]_j \parallel [\ ]_k$$

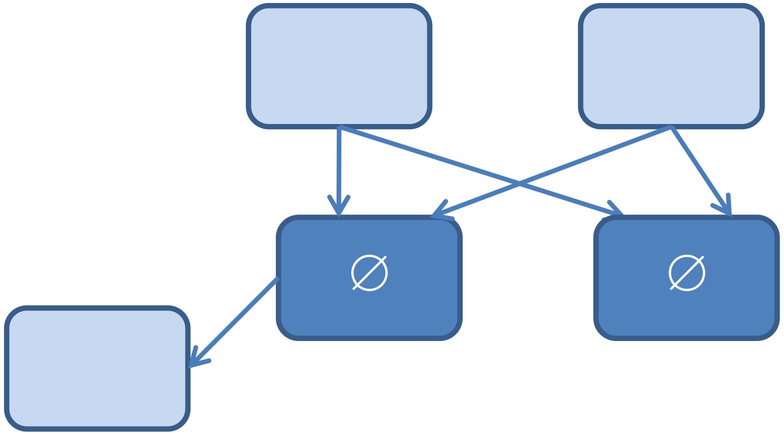
neuron division

E regular expression ‘test’

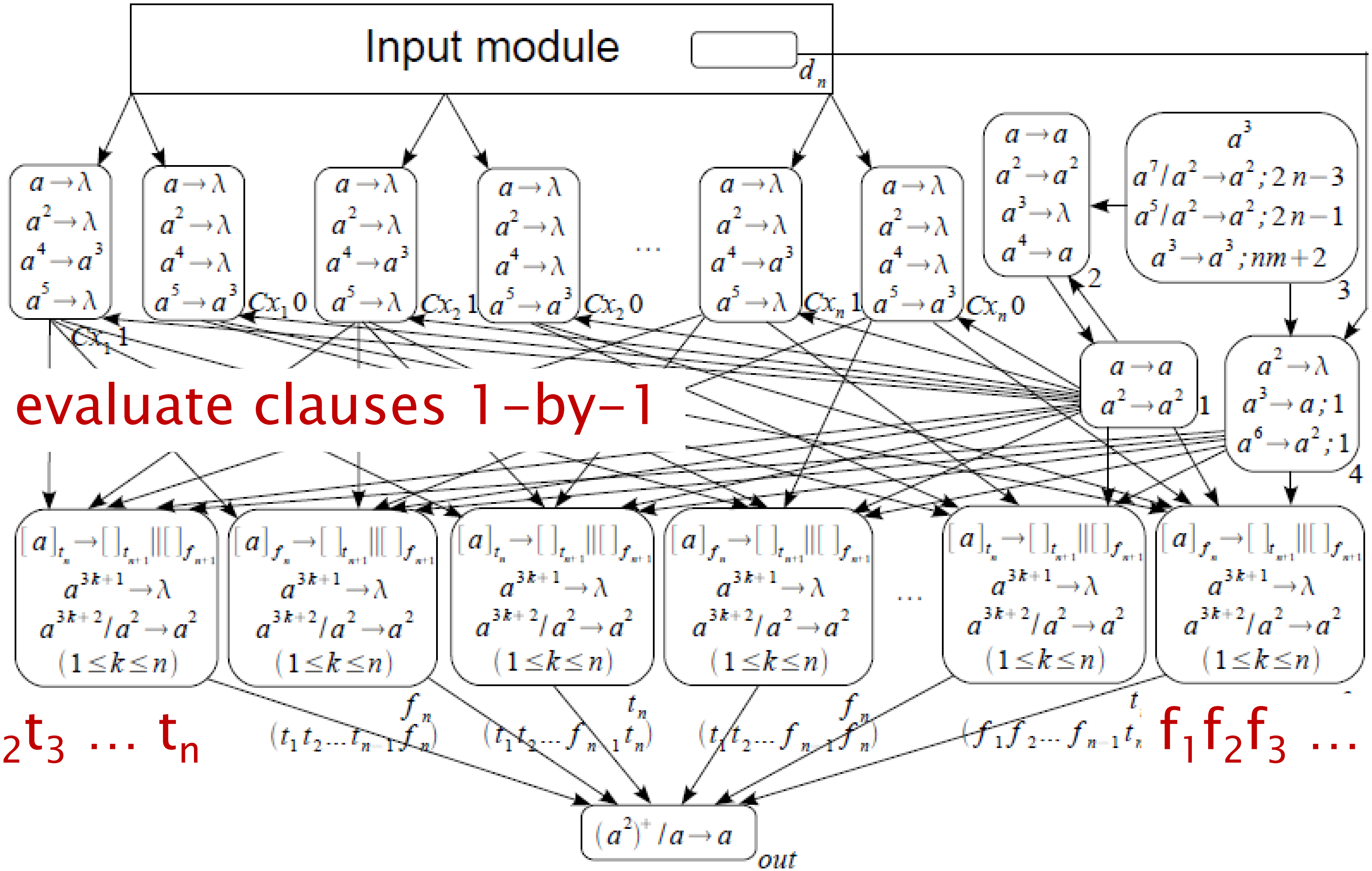
children inherit synapses
+ *synapse dictionary* (based on label)

no initial spikes





satisfiability and *output*



evaluate clauses 1–by–1

$t_1 t_2 t_3 \ldots t_n$ $f_1 f_2 f_3 \ldots f_n$

split when clause fails

nature computes!

how?
what?