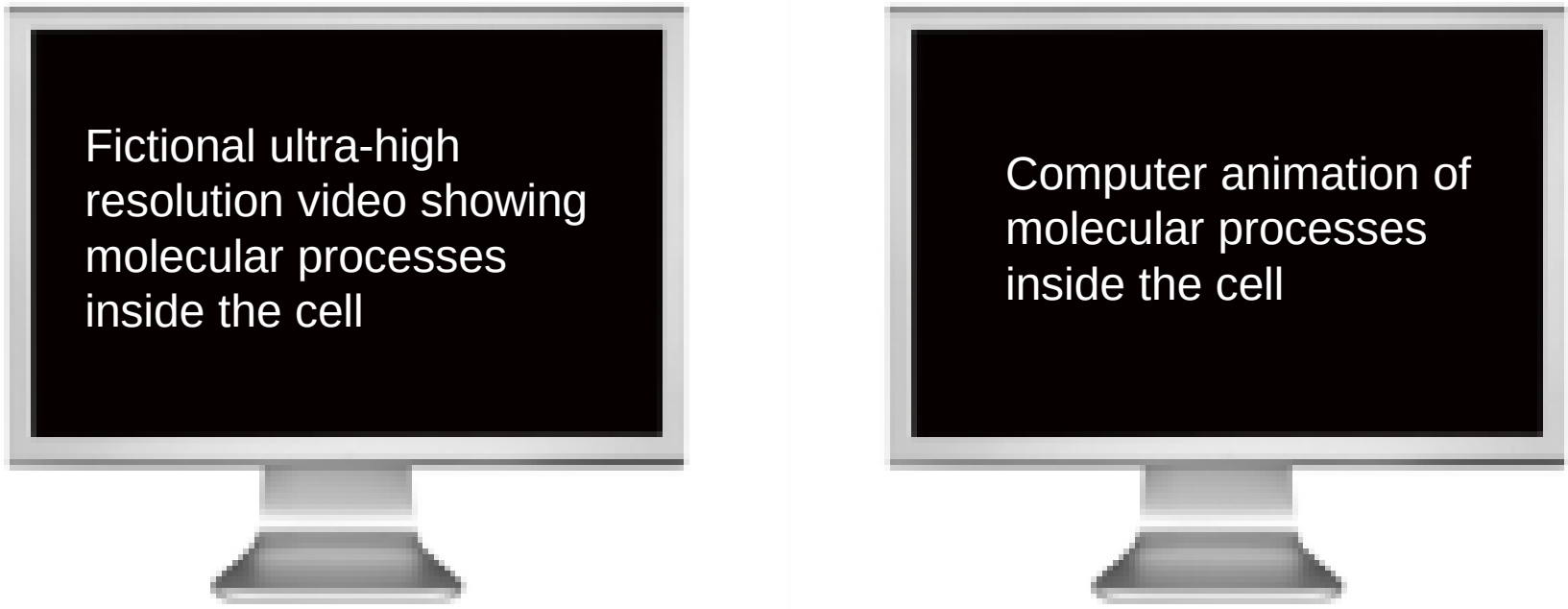
How to Build a Living Cell in Software
or
Can we computerize a bacterium?

Tom Henzinger

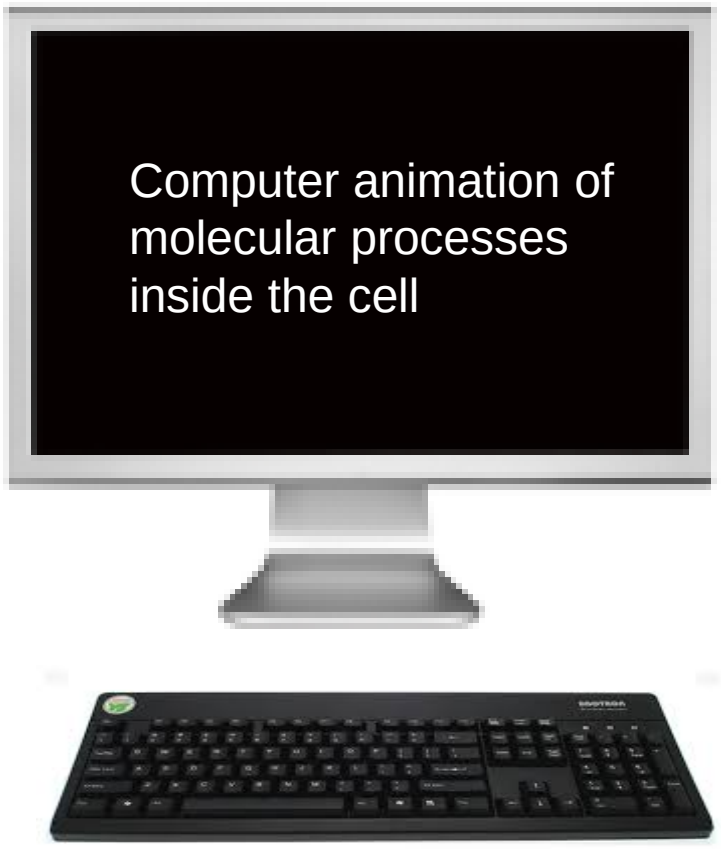
IST Austria

Turing Test for E. coli



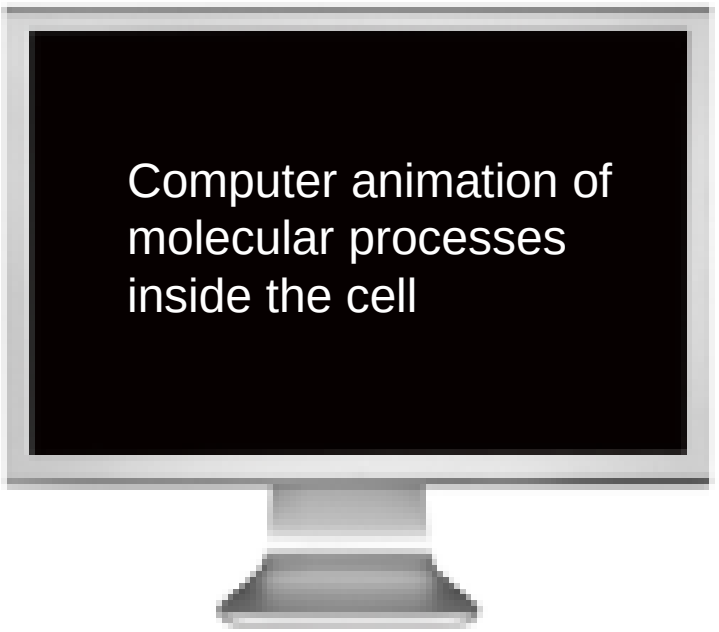
Biologist cannot distinguish.

Turing Test for E. coli



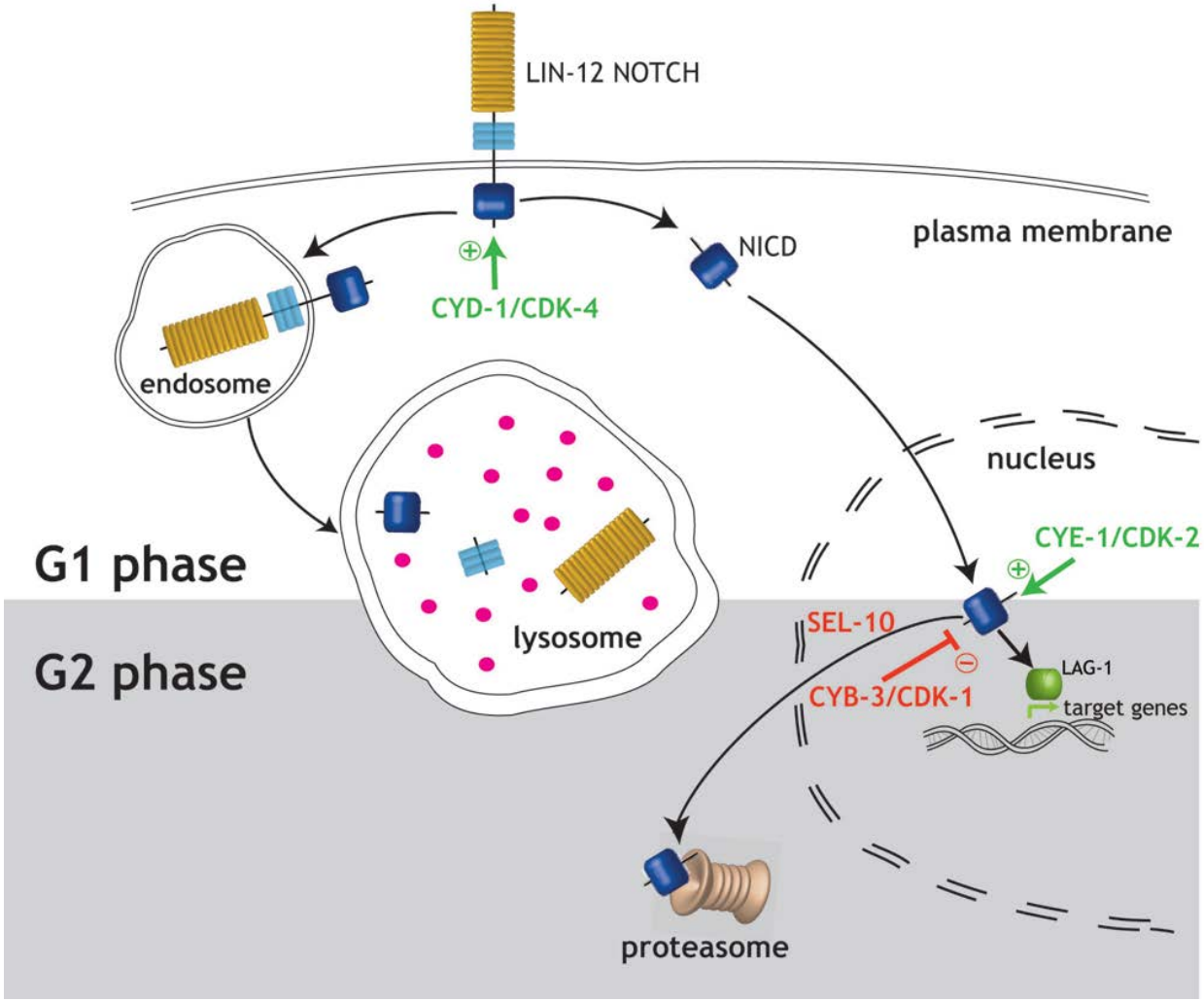
Biologist cannot distinguish.

Turing Test for E. coli

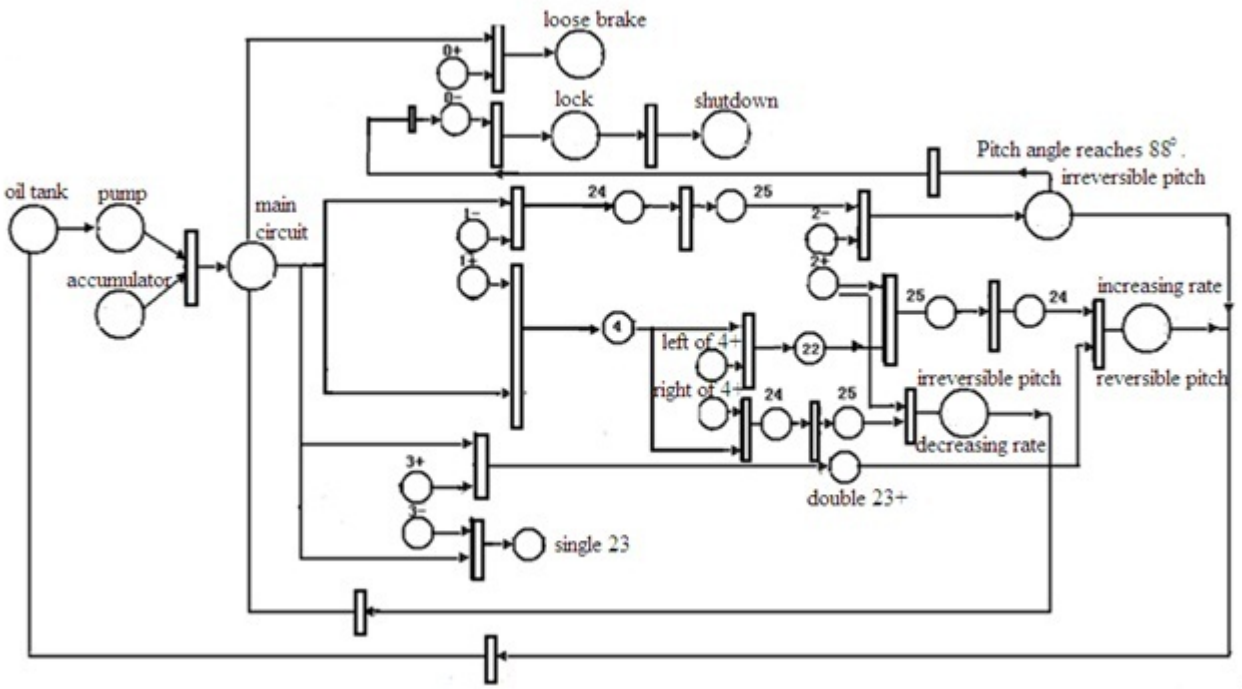


Such software must represent all actors and their interaction in real time (think “video game”).

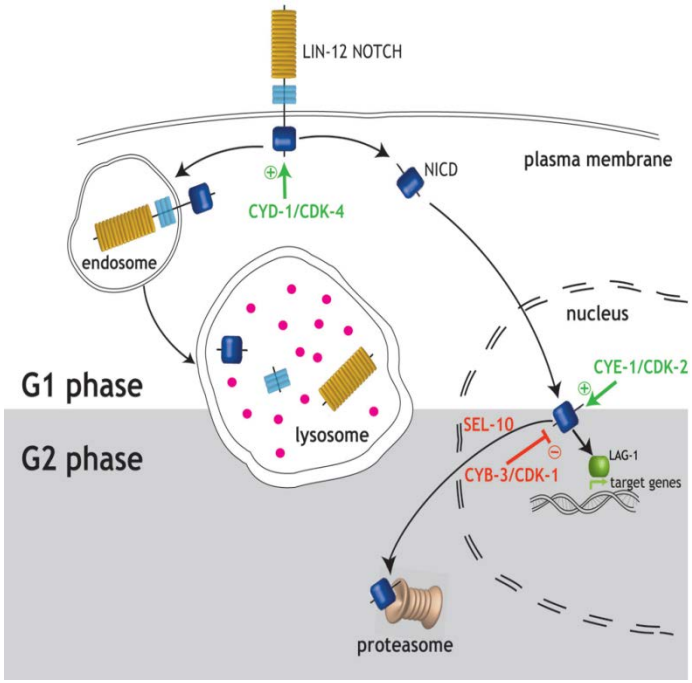
Cell cycle regulation of NOTCH signaling

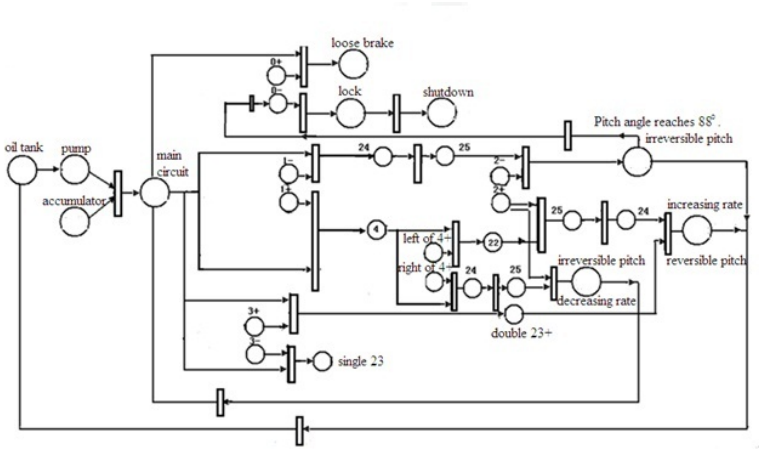


Nusser-Stein et al. 2012



Wind Turbine Hydraulic Variable Pitch System, Yang et al. Energies 2011





Which questions about the diagram can be answered with Yes or No?

The Cell as Reactive System [D. Harel]

Reacts to inputs:

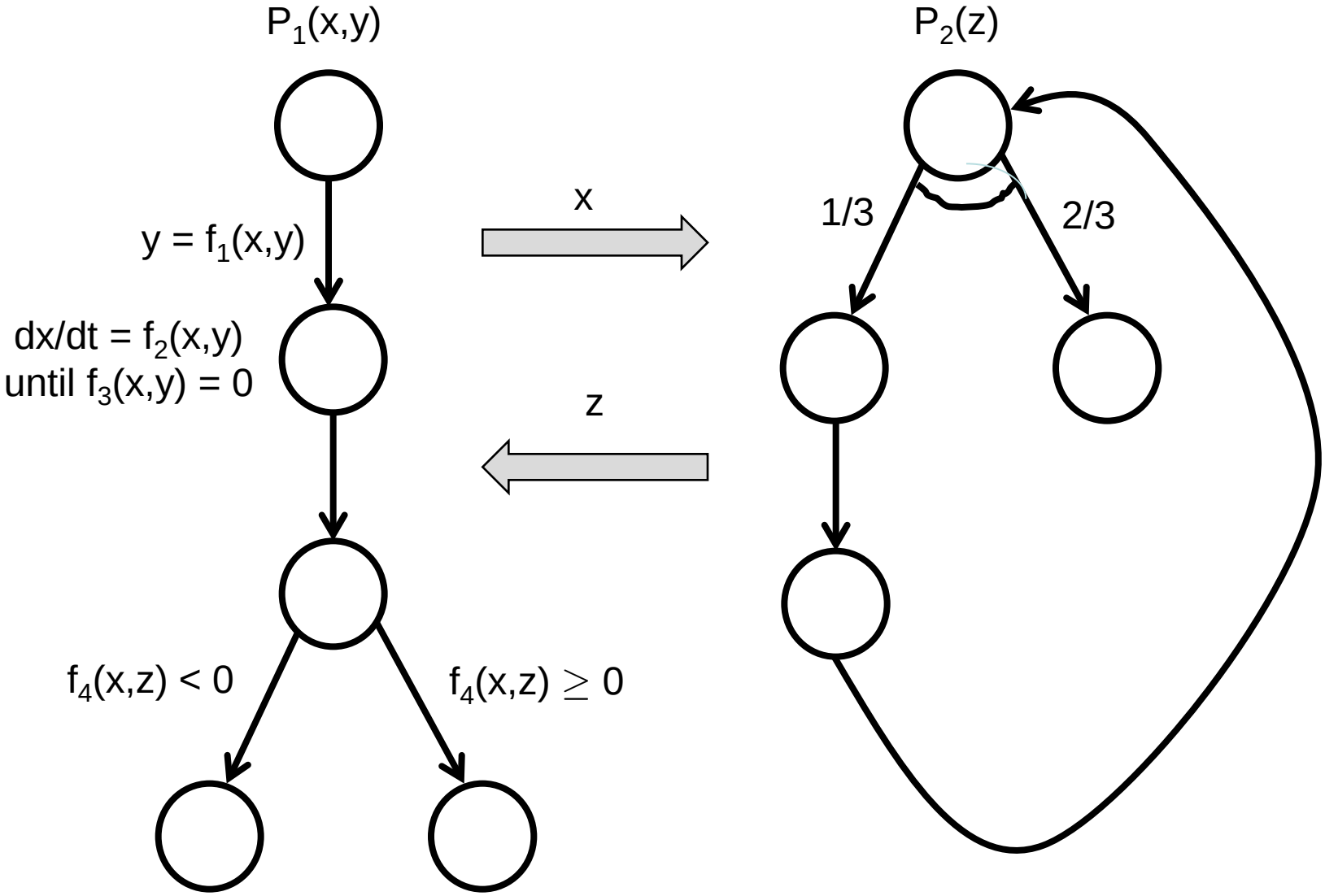
-physical (light, mechanical, electrical)
-chemical (molecular receptors, channels)

By producing observable outputs:

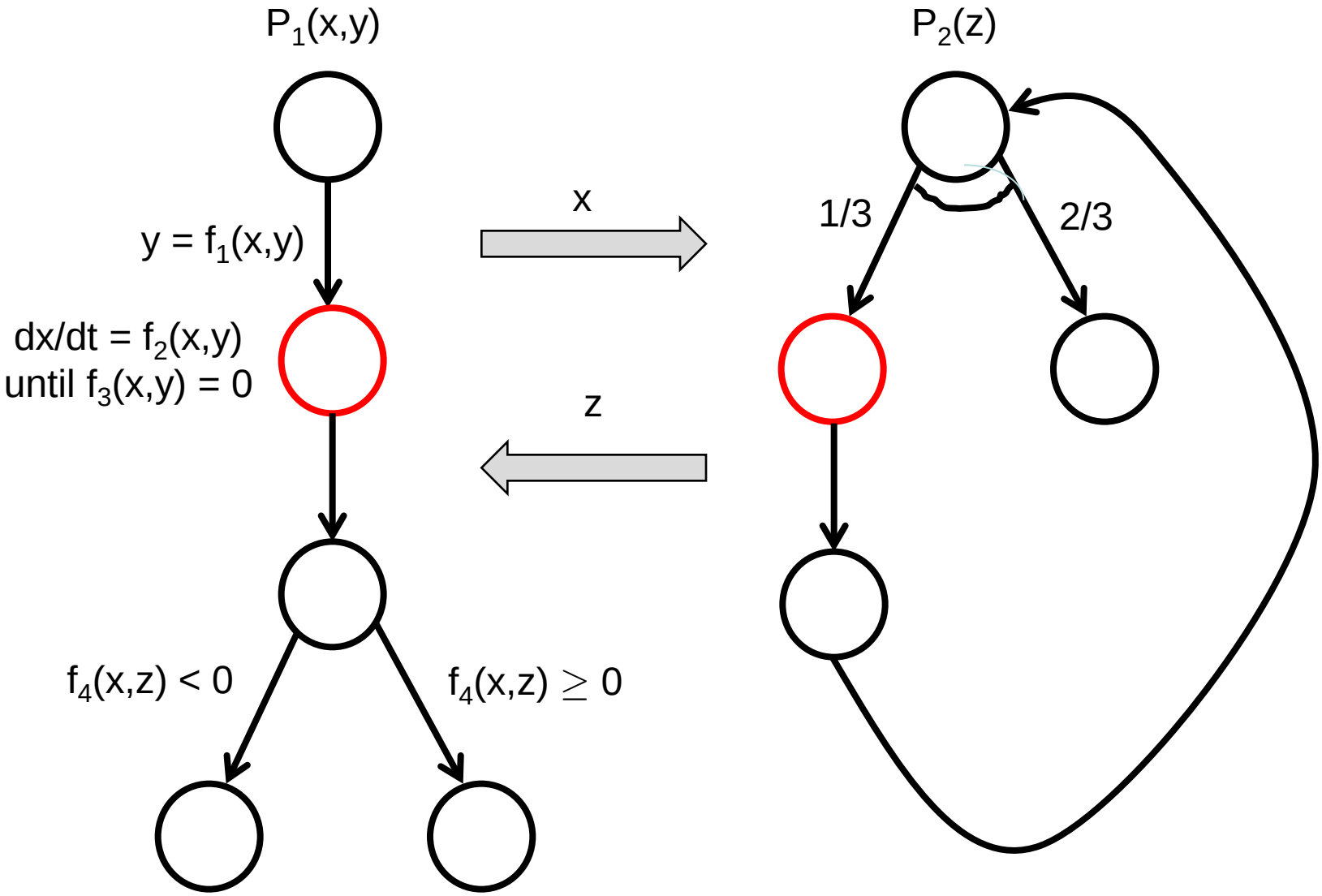
-growth, division, death
-movement
-signaling

Generated over time by an internal dynamics:

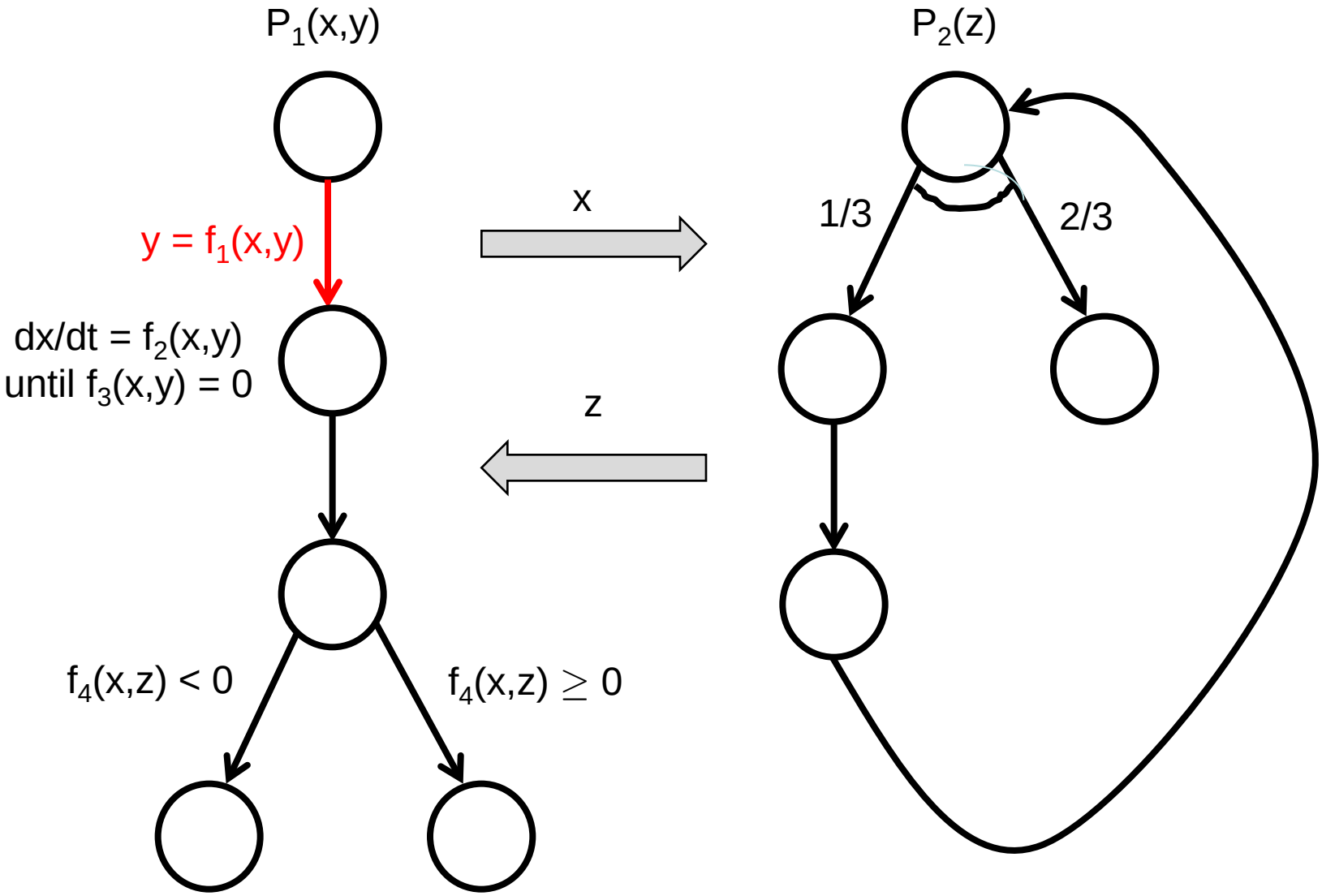
-metabolism
-cell fate
-cell cycle



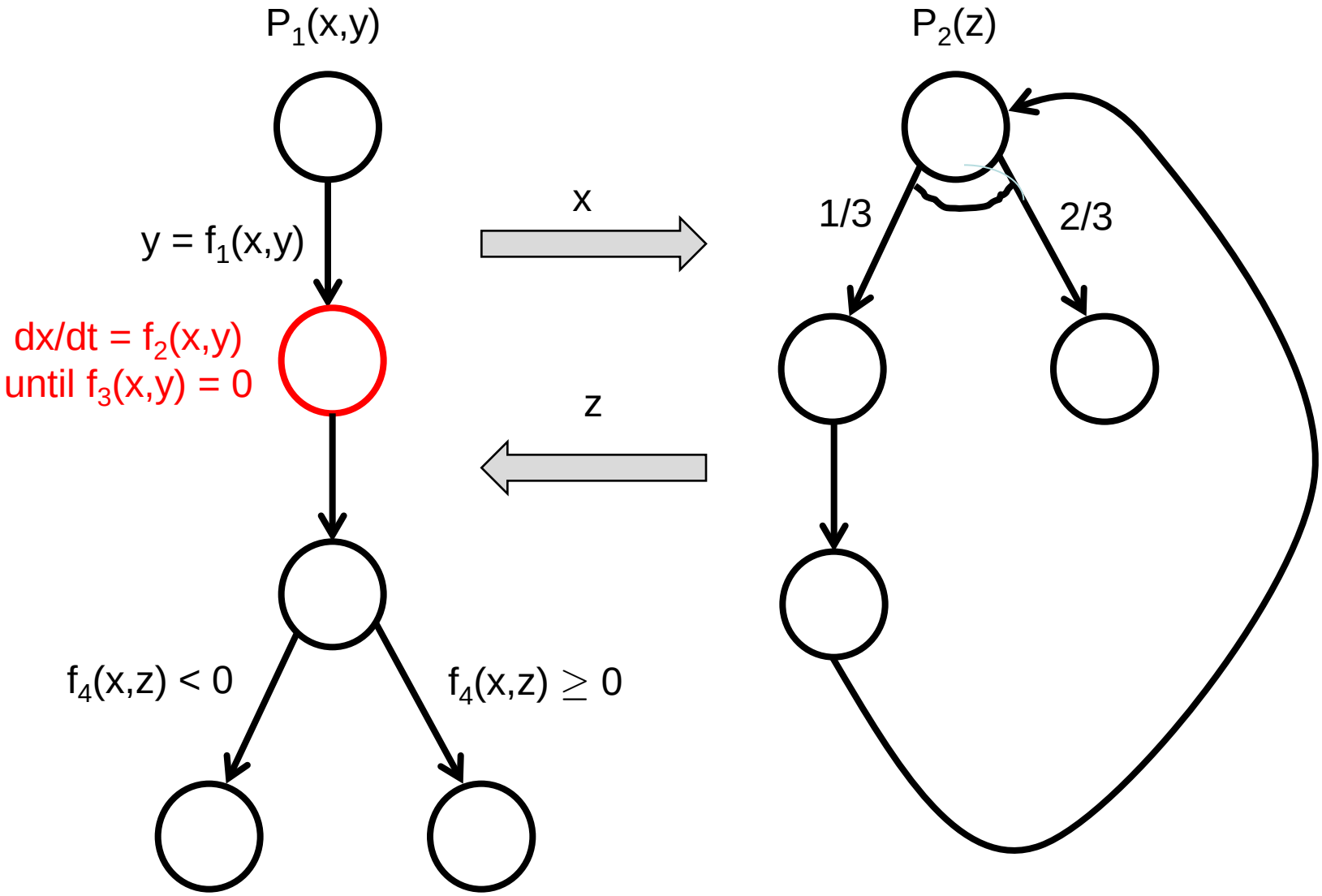
Reactive model



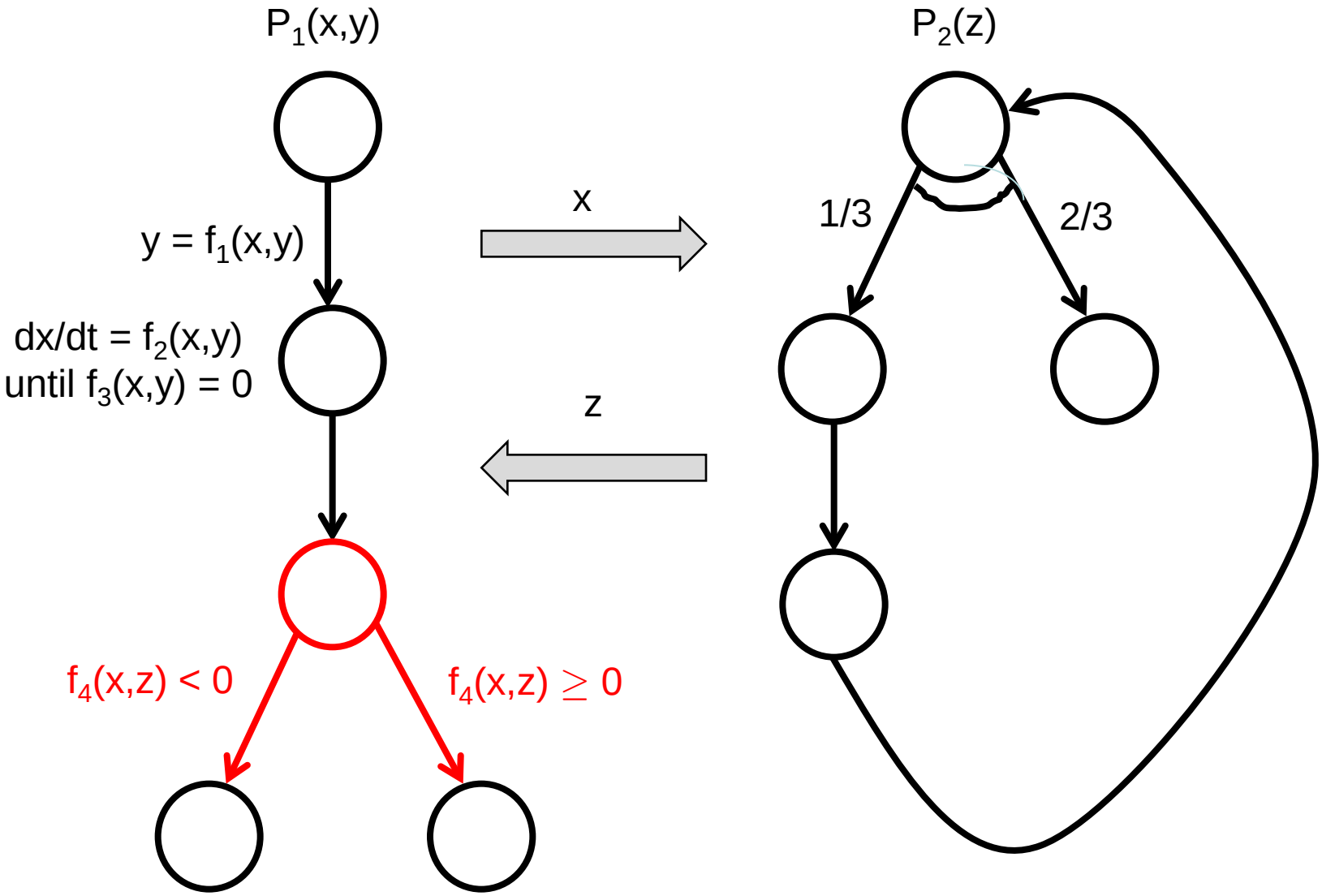
Distributed state



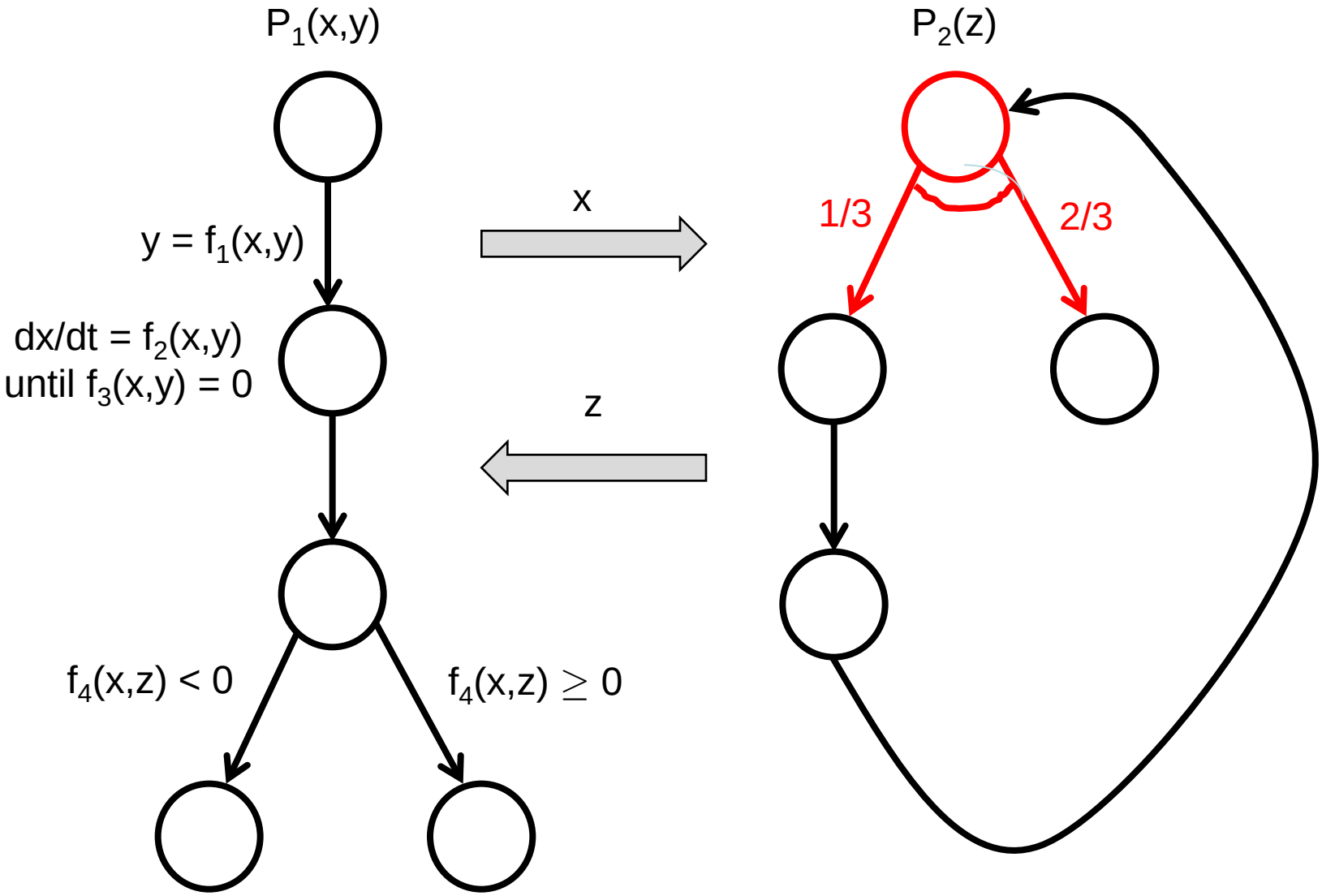
Discrete state change (Events)



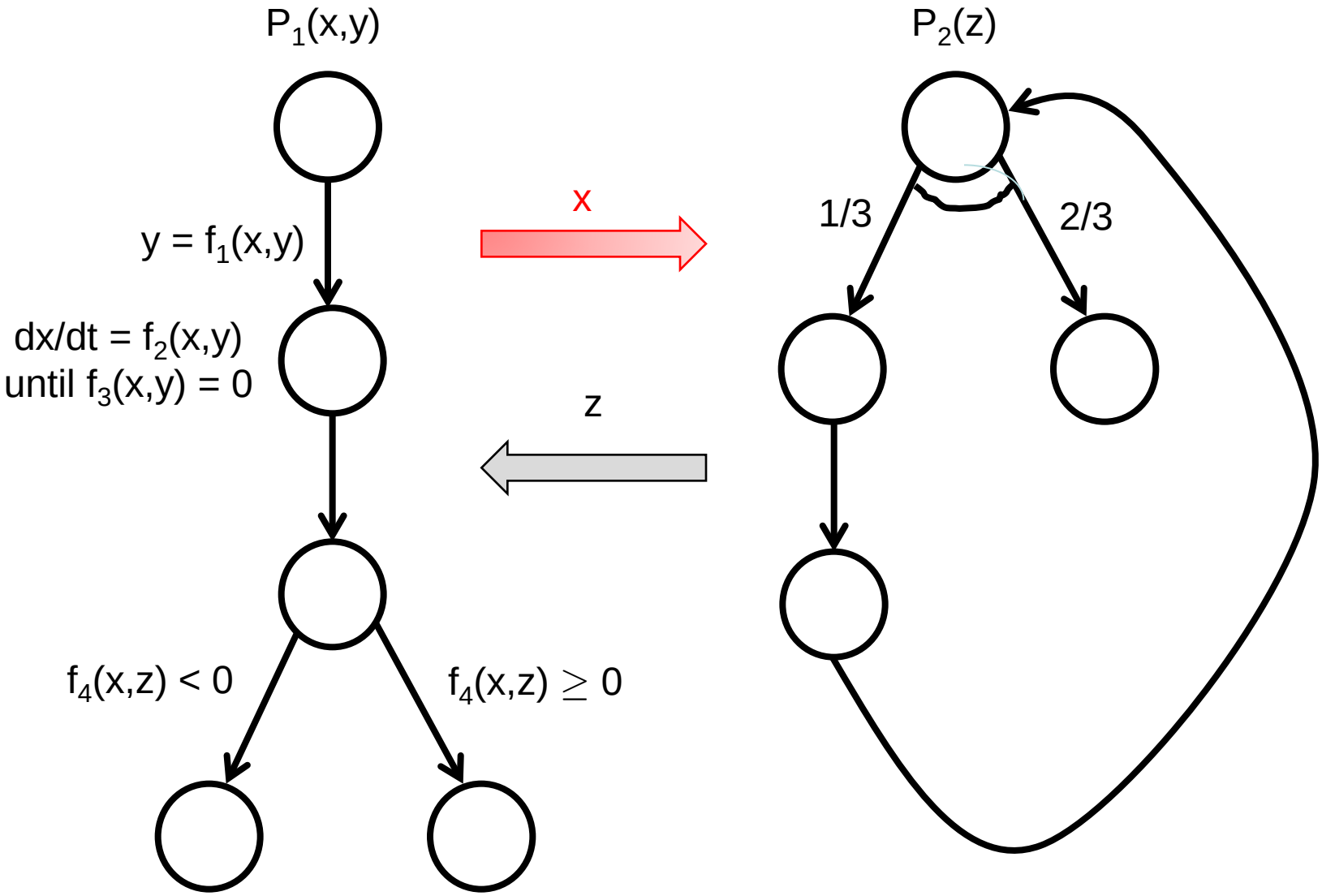
Continuous state change (Time)



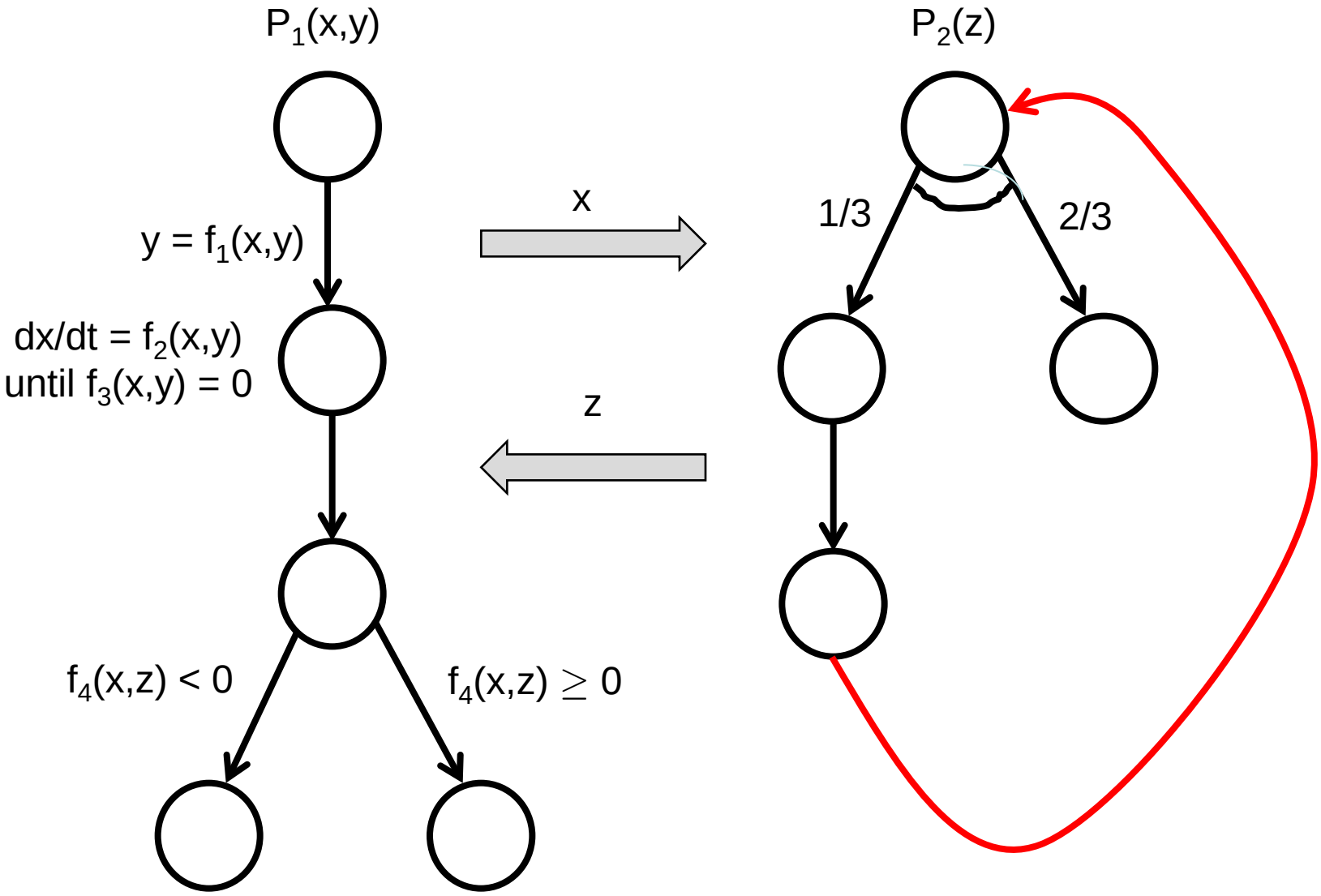
Decisions



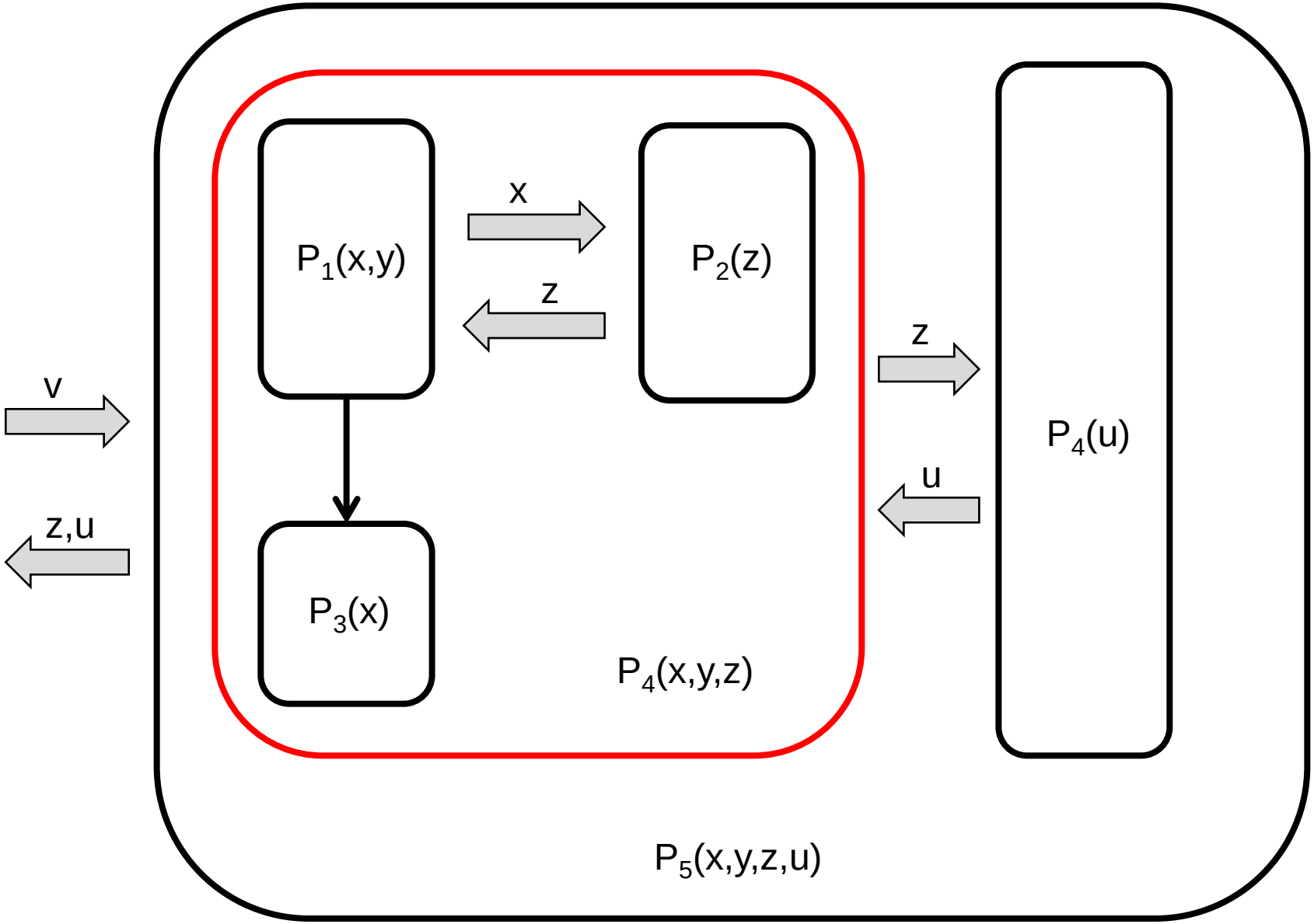
Randomness



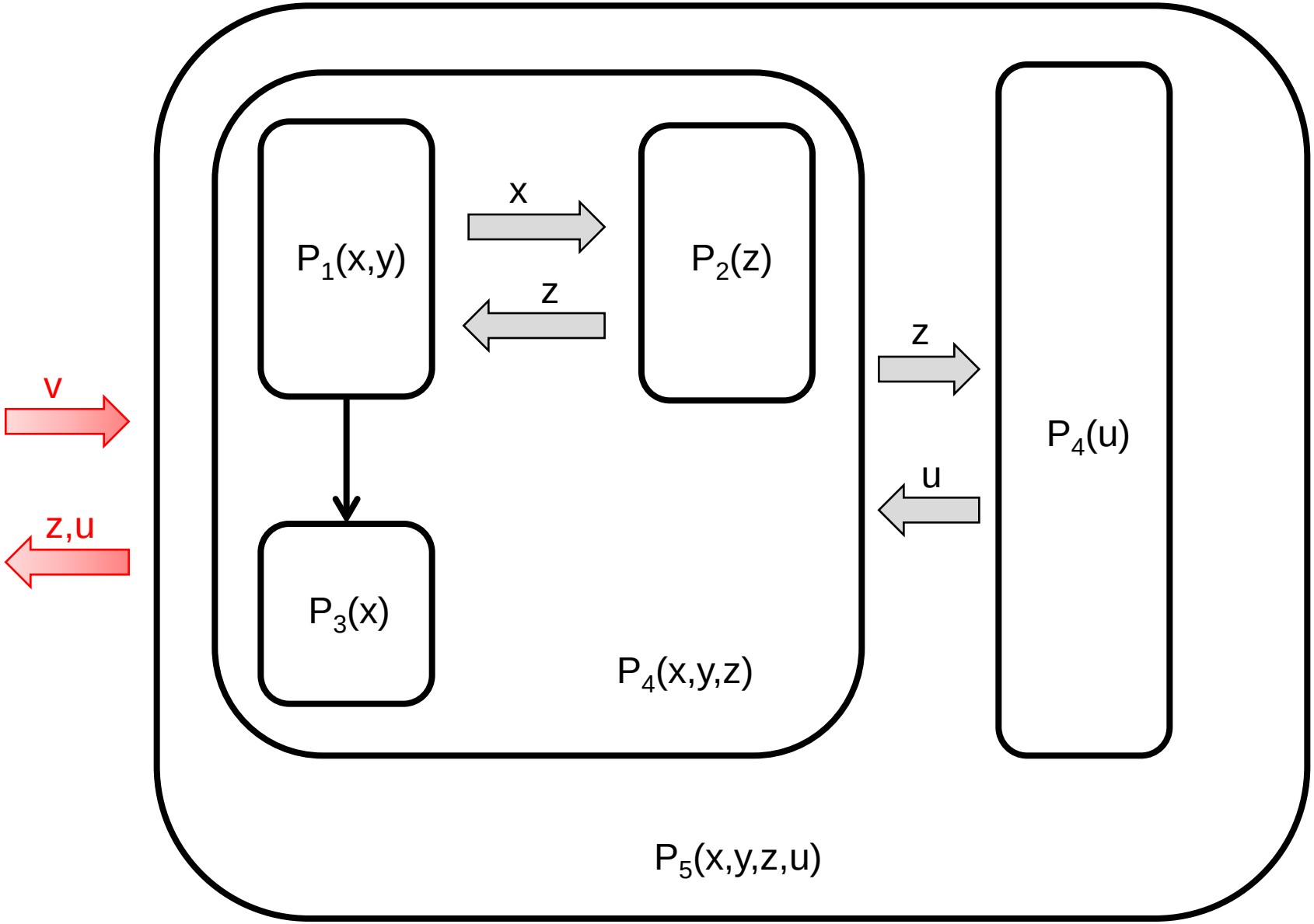
Interaction



Iteration



Hierarchy



Openness

Reactive Models are NOT particularly good for

-modeling space (proximity, polarity)

-approximate analysis

Standard Dynamic Models used in Molecular Biology

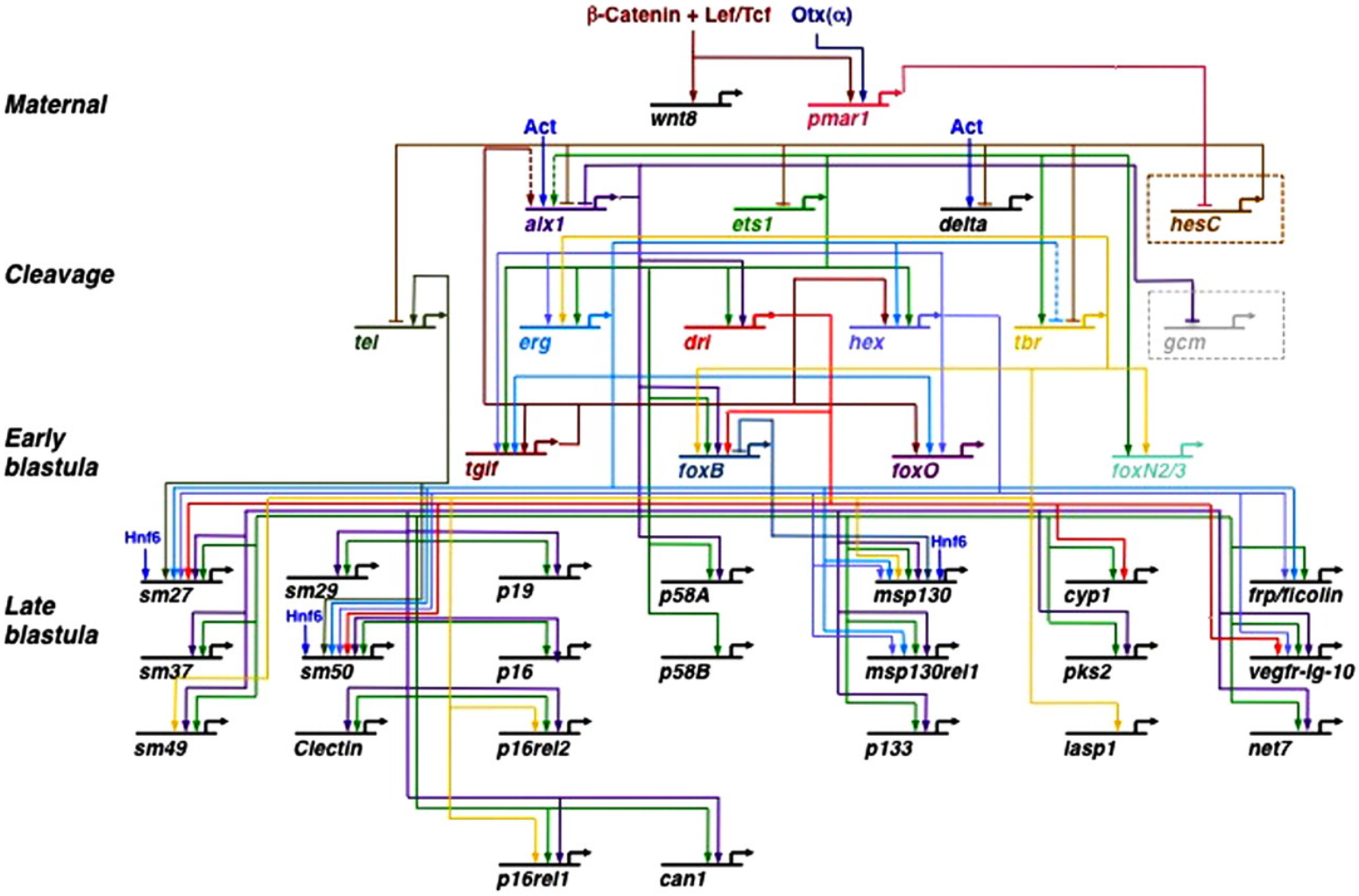
1 Markovian Population Models for chemical reaction systems
(continuous time, no decisions, no hierarchy)

2 Boolean/Qualitative Networks for activation/repression systems
(discrete time, no decisions, no hierarchy)

Chemical Reaction System (Phage λ)

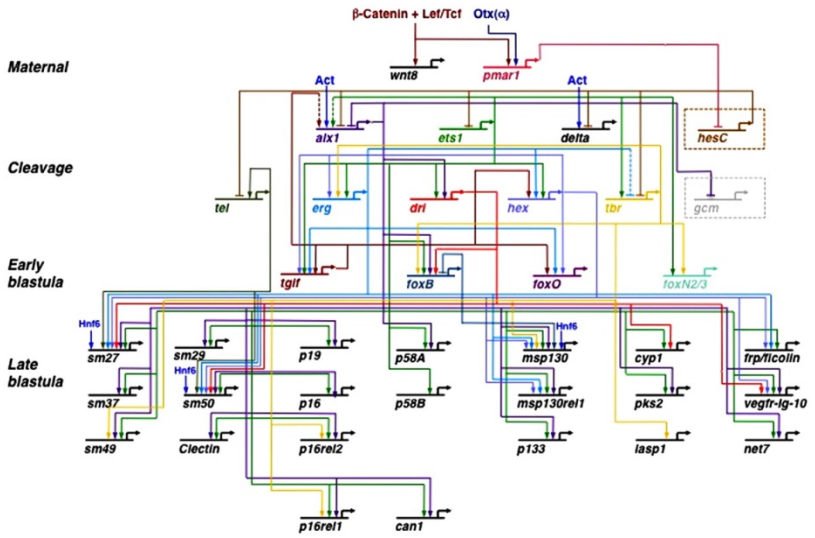
$\mathrm{RNA} \rightarrow \mathrm{RNA} + \mathrm{M}$	$c_1 = 0.043$
$\mathrm{M} \rightarrow \emptyset$	$c_2 = 0.0007$
$\mathrm{DNA.D} \rightarrow \mathrm{RNA}+\mathrm{DNA.D}$	$c_3 = 0.0715$
$\mathrm{RNA} \rightarrow \emptyset$	$c_4 = 0.0039$
$\mathrm{DNA} + \mathrm{D} \rightarrow \mathrm{DNA.D}$	$c_5 = 0.0199$
$\mathrm{DNA.D} \rightarrow \mathrm{DNA} + \mathrm{D}$	$c_6 = 0.479$
$\mathrm{DNA.D} + \mathrm{D} \rightarrow \mathrm{DNA.2D}$	$c_7 = 0.0002$
$\mathrm{DNA.2D} \rightarrow \mathrm{DNA.D} + \mathrm{D}$	$c_8 = 9 \times 10^{-12}$
$\mathrm{M} + \mathrm{M} \rightarrow \mathrm{D}$	$c_9 = 0.083$
$\mathrm{D} \rightarrow \mathrm{M} + \mathrm{M}$	$c_{10} = 0.5$

Gene Activation/Repression System



Rafiq K et al. Development 2012; 139:579-590.

$$
\begin{aligned}
&\mathrm{RNA} \rightarrow \mathrm{RNA} + \mathrm{M} && c_1 = 0.043 \\
&\mathrm{M} \rightarrow \emptyset && c_2 = 0.0007 \\
&\mathrm{DNA.D} \rightarrow \mathrm{RNA}{+}\mathrm{DNA.D} && c_3 = 0.0715 \\
&\mathrm{RNA} \rightarrow \emptyset && c_4 = 0.0039 \\
&\mathrm{DNA} + \mathrm{D} \rightarrow \mathrm{DNA.D} && c_5 = 0.0199 \\
&\mathrm{DNA.D} \rightarrow \mathrm{DNA} + \mathrm{D} && c_6 = 0.479 \\
&\mathrm{DNA.D} + \mathrm{D} \rightarrow \mathrm{DNA.2D} && c_7 = 0.0002 \\
&\mathrm{DNA.2D} \rightarrow \mathrm{DNA.D} + \mathrm{D} && c_8 = 9 \times 10^{-12} \\
&\mathrm{M} + \mathrm{M} \rightarrow \mathrm{D} && c_9 = 0.083 \\
&\mathrm{D} \rightarrow \mathrm{M} + \mathrm{M} && c_{10} = 0.5
\end{aligned}
$$

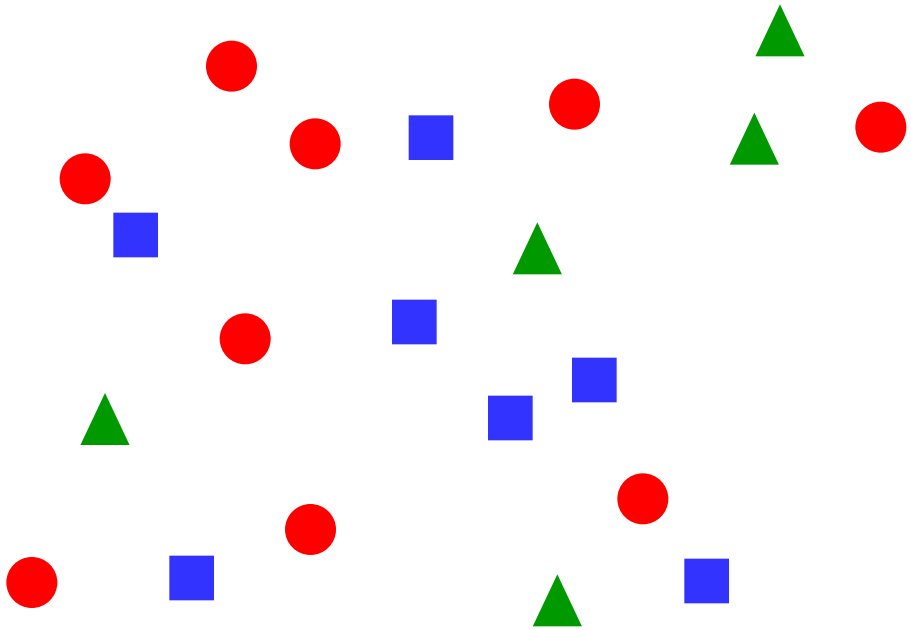


Continuous time, stochastic

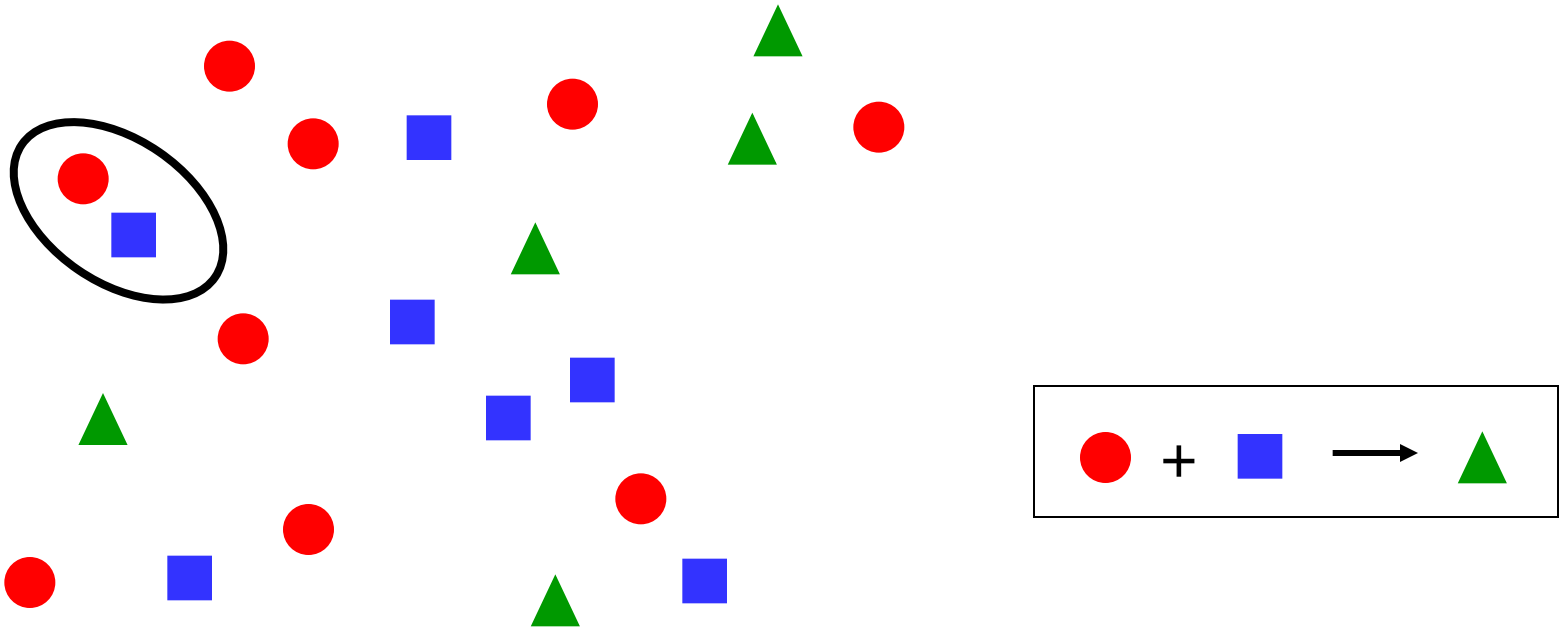
Discrete time, deterministic

No decisions, no hierarchy

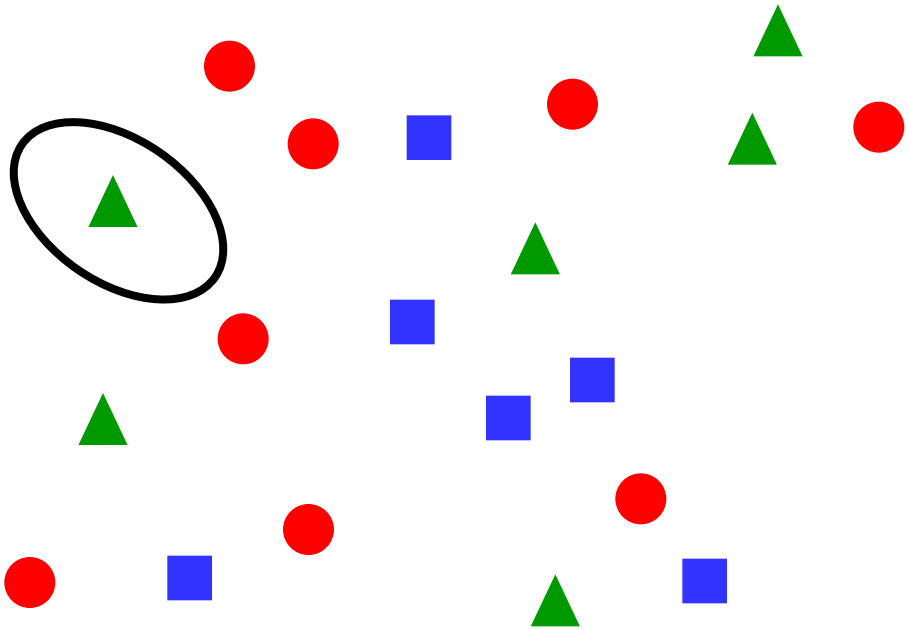
Markovian Population Models

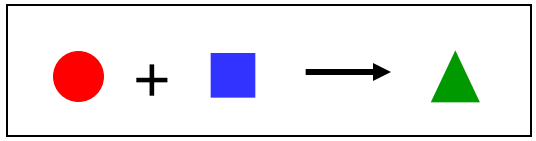


Markovian Population Models

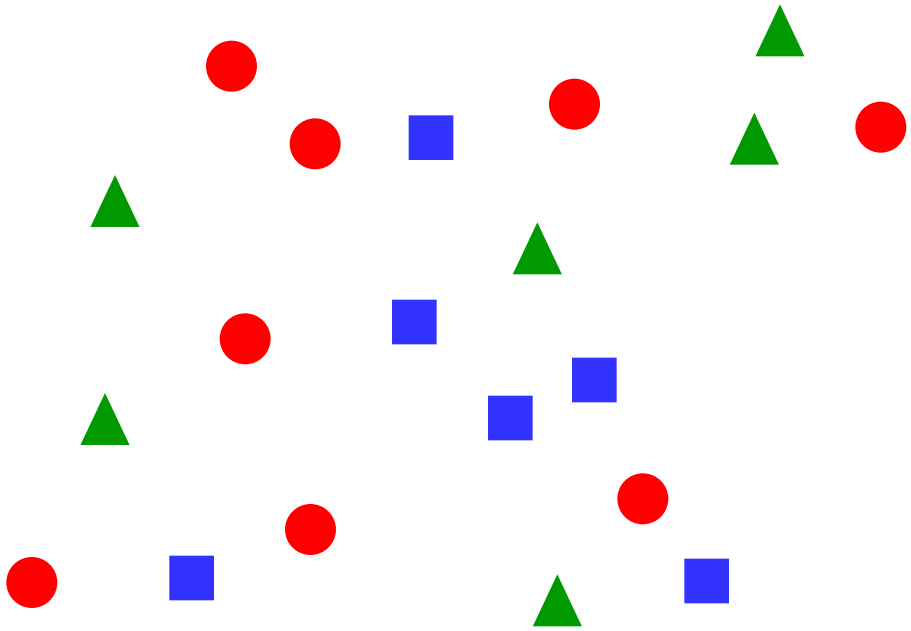


Markovian Population Models



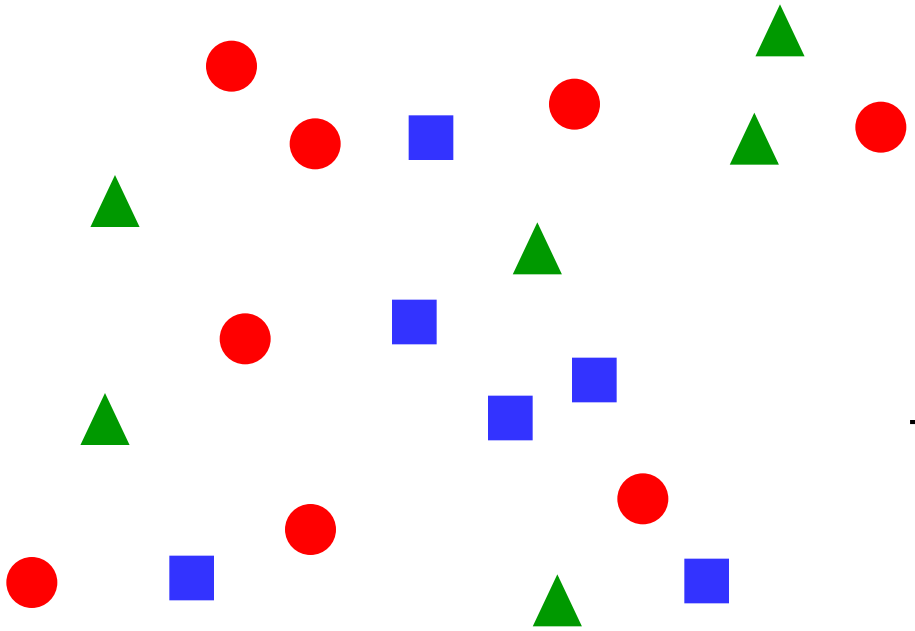


Markovian Population Models



State: (8 , 6 , 6)

Markovian Population Models

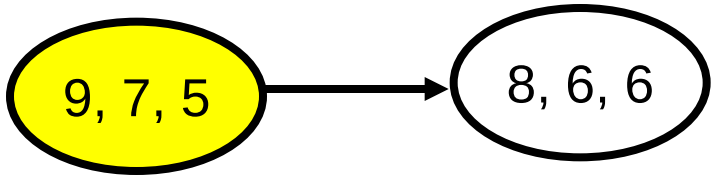


(9 , 7 , 5)

Transition:

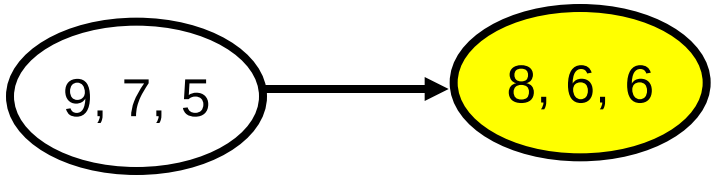
(8 , 6 , 6)

State Transition Systems



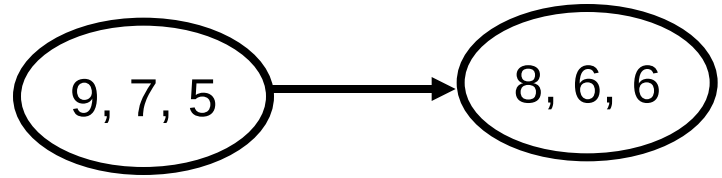
deterministic

State Transition Systems

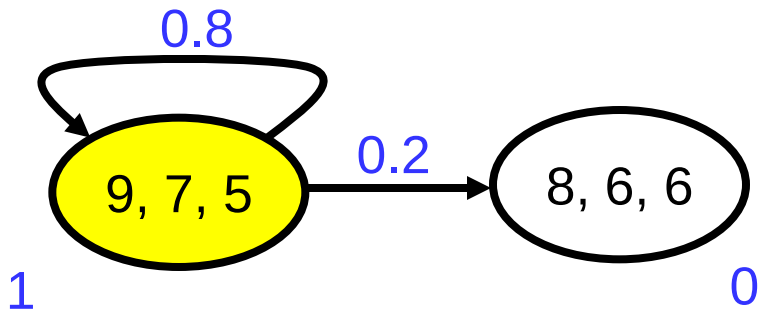


deterministic

State Transition Systems

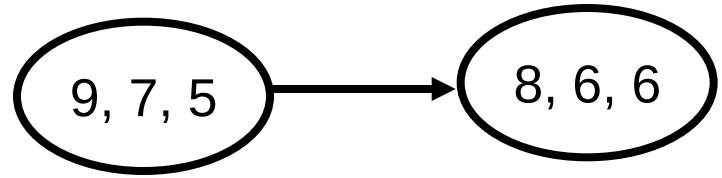


deterministic

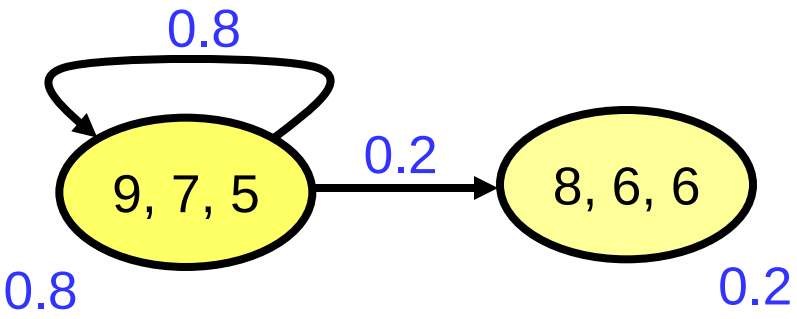


discrete stochastic
time 0

State Transition Systems

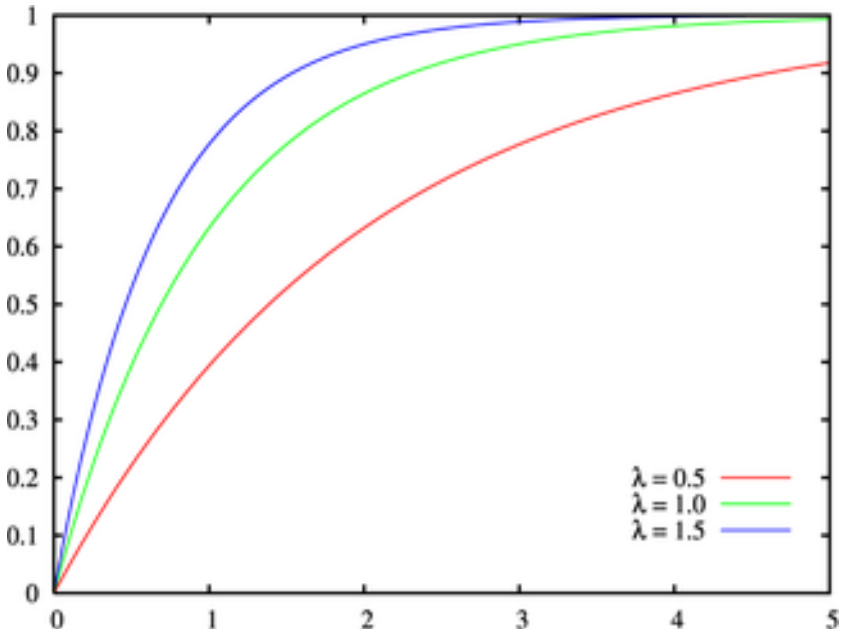


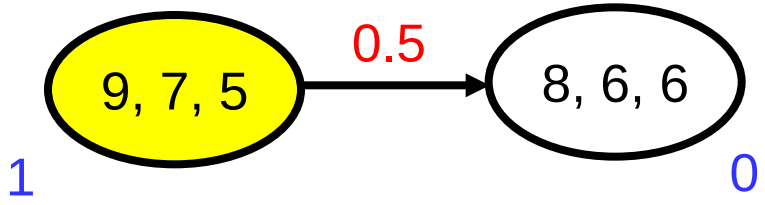
deterministic



discrete stochastic
time 1

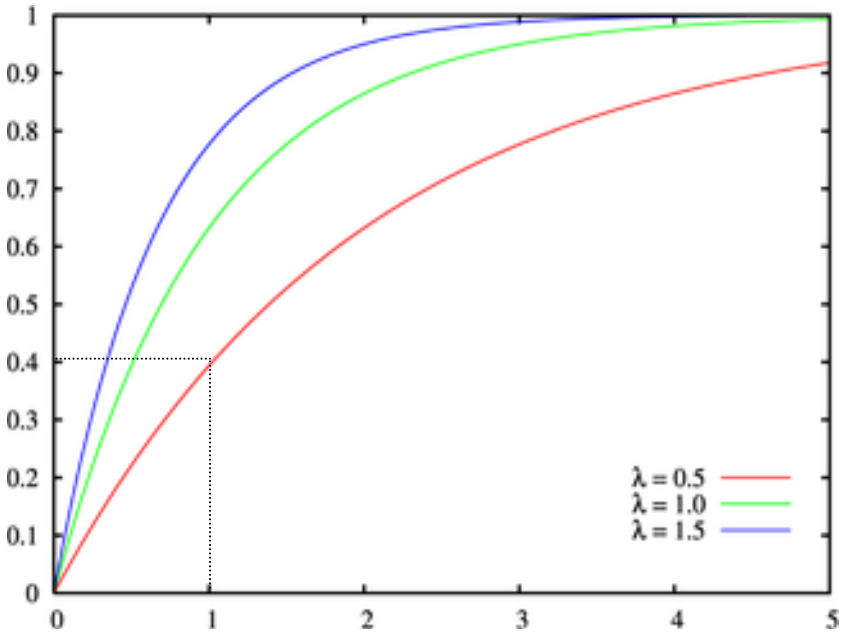
State Transition Systems

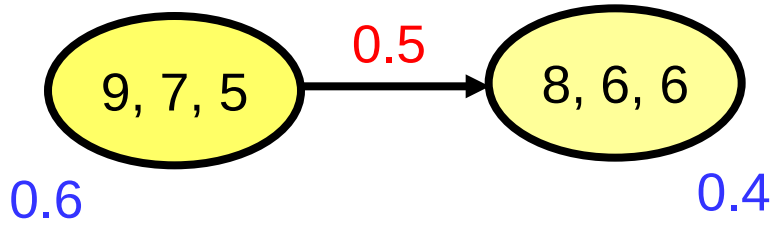




continuous stochastic
time 0
exit rate 0.5
expected residence time 2

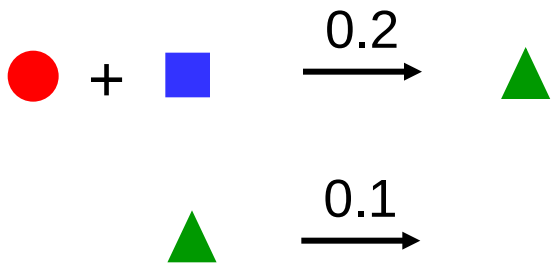
State Transition Systems



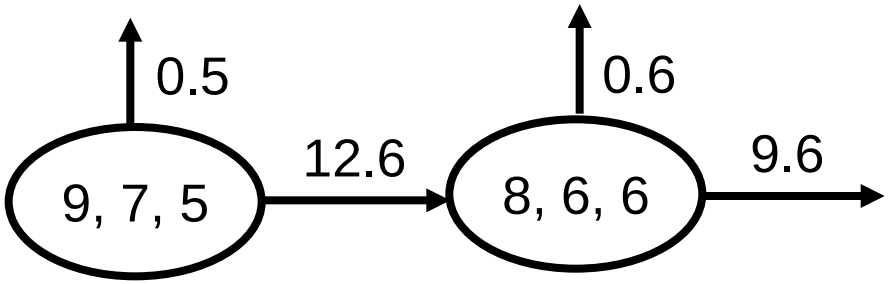


continuous stochastic
time 1
exit rate 0.5
expected residence time 2

Markovian Population Models

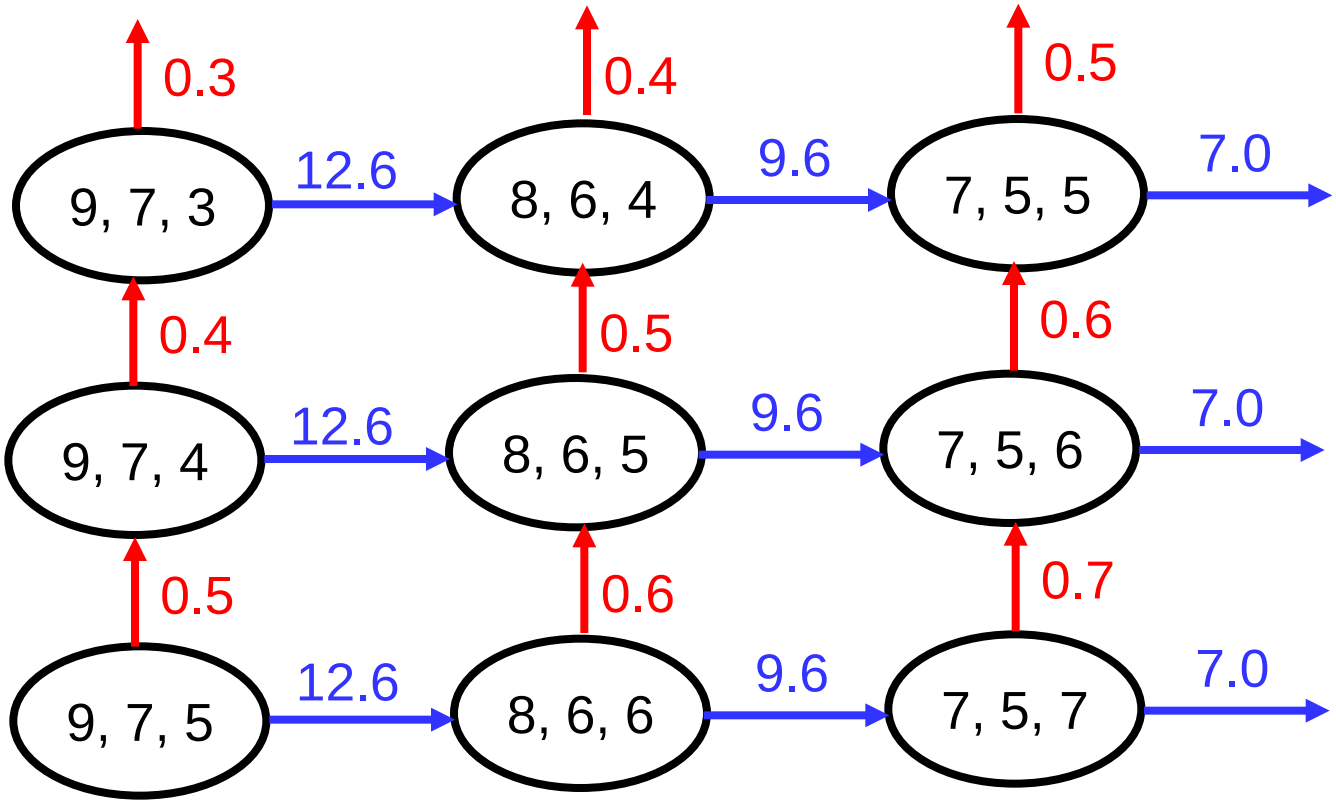


Syntax: stoichiometric equations (finite object)

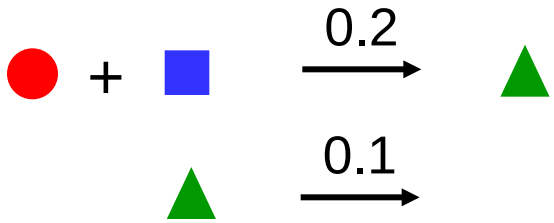


Semantics: continuous-time Markov chain (infinite object)

Continuous-Time Markov Chain (CTMC)



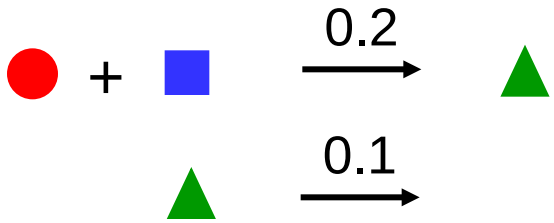
Chemist's syntax:



Physicist's syntax (chemical master equation):

$$\delta p^t(x) / \delta t = \Sigma_{i:\, x \in H_i} \alpha_i(u_i^{-1}(x)) \cdot p^t(u_i^{-1}(x)) - \Sigma_{i:\, x \in G_i} \alpha_i(x) \cdot p^t(x)$$

Chemist's syntax:



Physicist's syntax (chemical master equation):

$$\delta p^t(x) / \delta t = \Sigma_{i:\, x \in H_i} \alpha_i(u_i^{-1}(x)) \cdot p^t(u_i^{-1}(x)) - \Sigma_{i:\, x \in G_i} \alpha_i(x) \cdot p^t(x)$$

Computer scientist's syntax (stochastic reactive module):

$$[]\ x_1 \geq 1 \wedge x_2 \geq 1 \quad -0.2 \cdot x_1 \cdot x_2 \rightarrow \quad x_1 := x_1 - 1;\ x_2 := x_2 - 1;\ x_3 := x_3 + 1$$
$$[]\ x_3 \geq 1 \quad -0.1 \cdot x_3 \rightarrow \quad x_3 := x_3 - 1$$

Syntax Matters

1. Expressiveness	0
2. Succinctness	𝍸 𝍸 IIII
3. Operations	+1 addition multiplication

XIV x XXXIV vs. 14 x 34 =

$$\begin{array}{r} 42 \\ 56 \\ \hline 476 \end{array}$$

Executable Biology

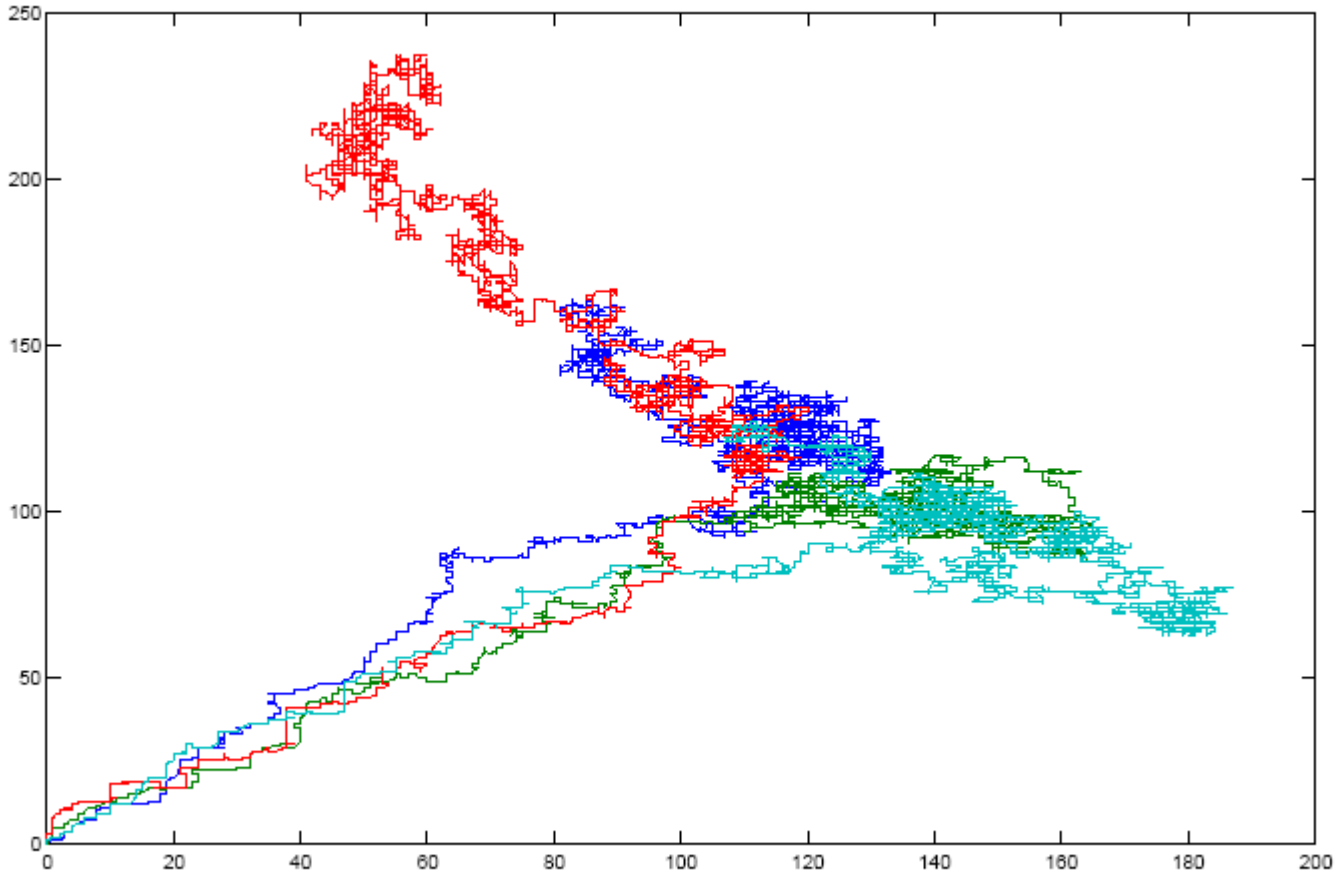
"Machine model" (rather than equational model):

[] $x_1 \geq 1 \wedge x_2 \geq 1$ - $0.2 \cdot x_1 \cdot x_2$ -> $x_1 := x_1 - 1$; $x_2 := x_2 - 1$; $x_3 := x_3 + 1$
[] $x_3 \geq 1$ - $0.1 \cdot x_3$ -> $x_3 := x_3 - 1$

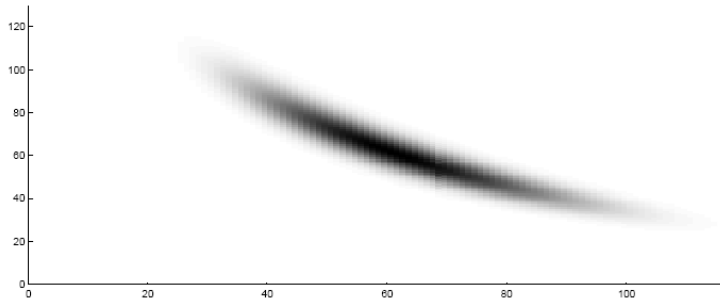
-can be executed (not "simulated")
-can be composed, encapsulated, and abstracted
-can be dynamically reconfigured
-can be "formally verified" (if you are lucky)

Transient behavior (rather than steady state) of interest.

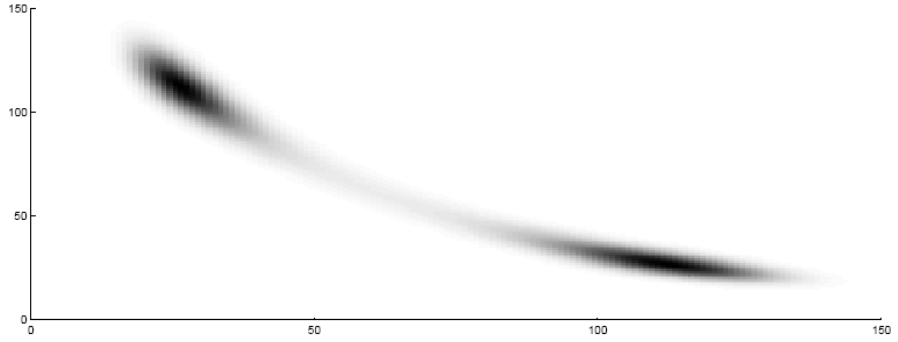
Four Executions for Phage λ Genetic Switch



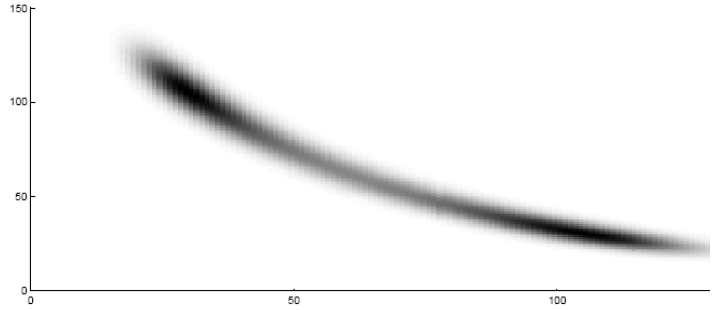
Probability Mass Propagation for Phage λ Genetic Switch



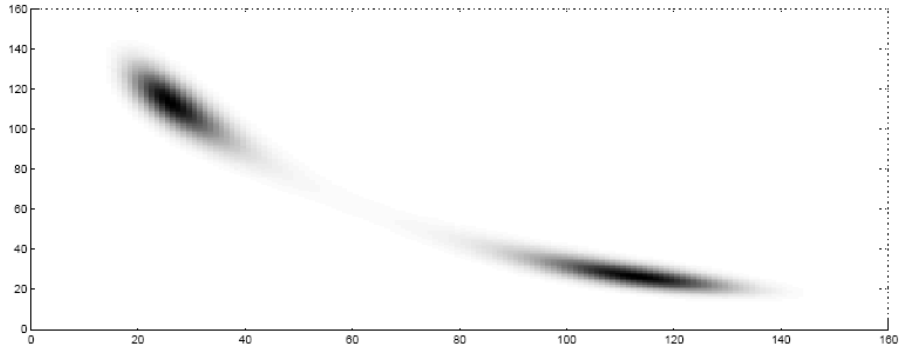
t = 5000



t = 30000

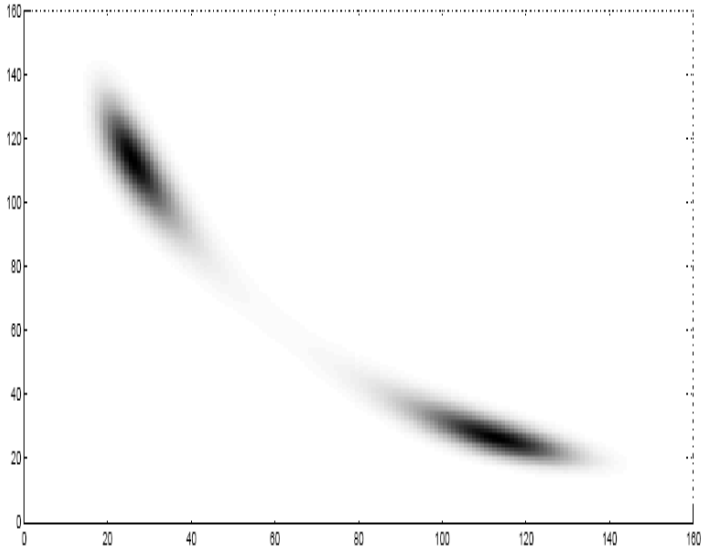


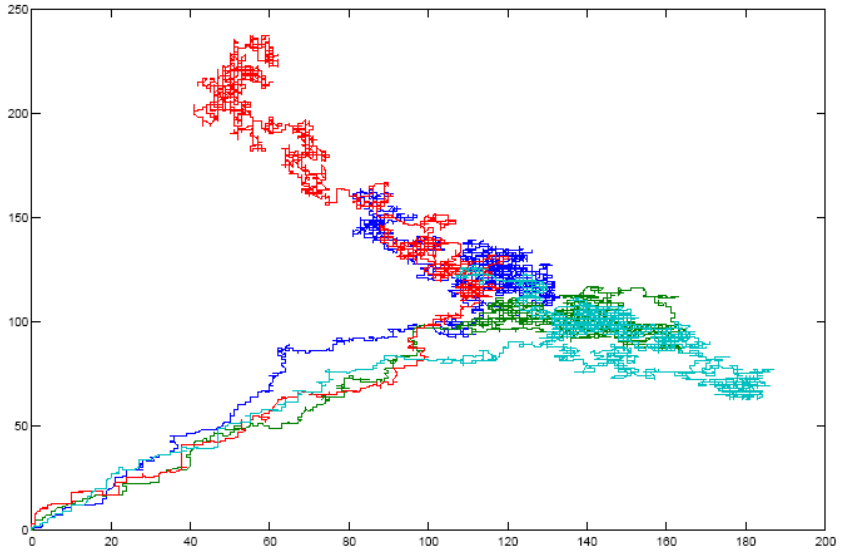
t = 15000



t = 50000

Mass Propagation vs. Execution





Program verification vs. Program testing

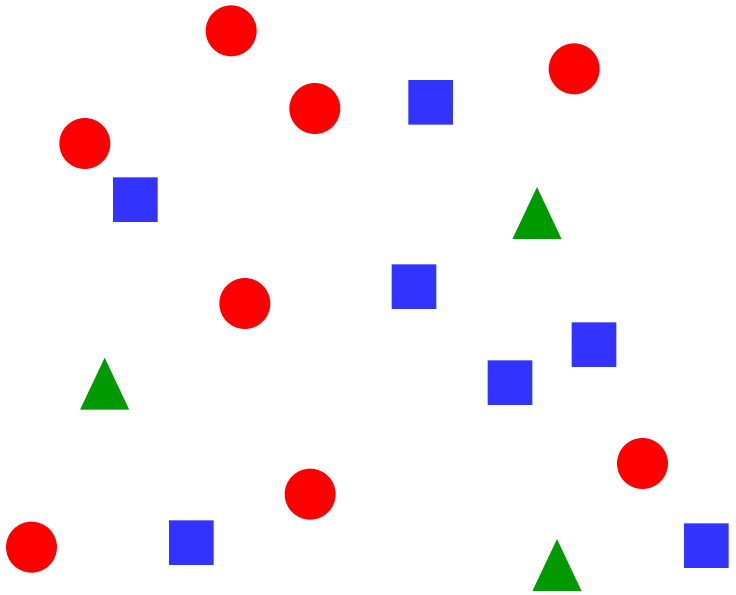
Phage λ Model

Desired precision: 3×10^{-6}

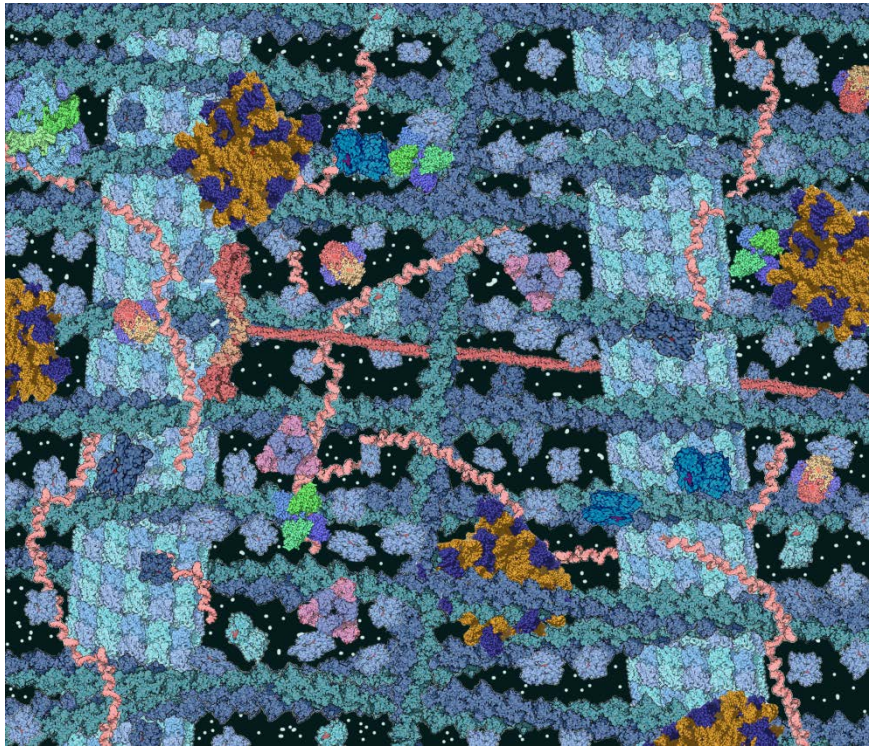
Probability mass propagation: 55 min runtime
(adaptive uniformization with sliding-window abstraction)
[Didier, H., Mateescu, Wolf 2011]

Execution ($\beta = 0.95$): 67 h runtime (3×10^8 runs)

Main Problem with Biochemical Reaction Systems



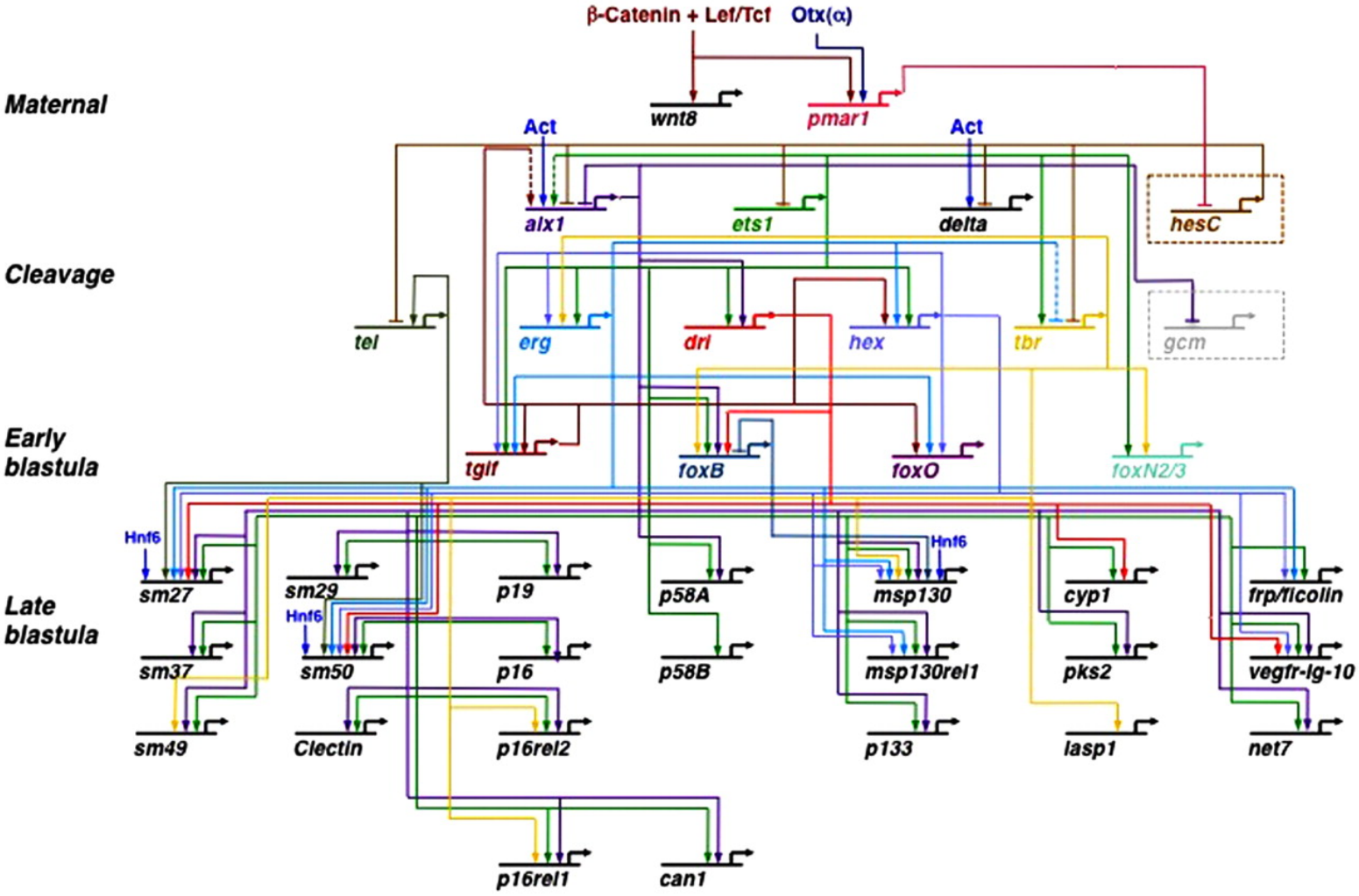
Well-stirred mixture (gas or fluid)



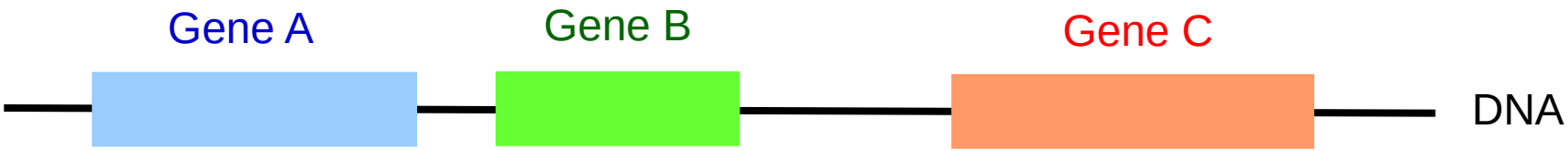
Crowded cytosol [T. Vickers]

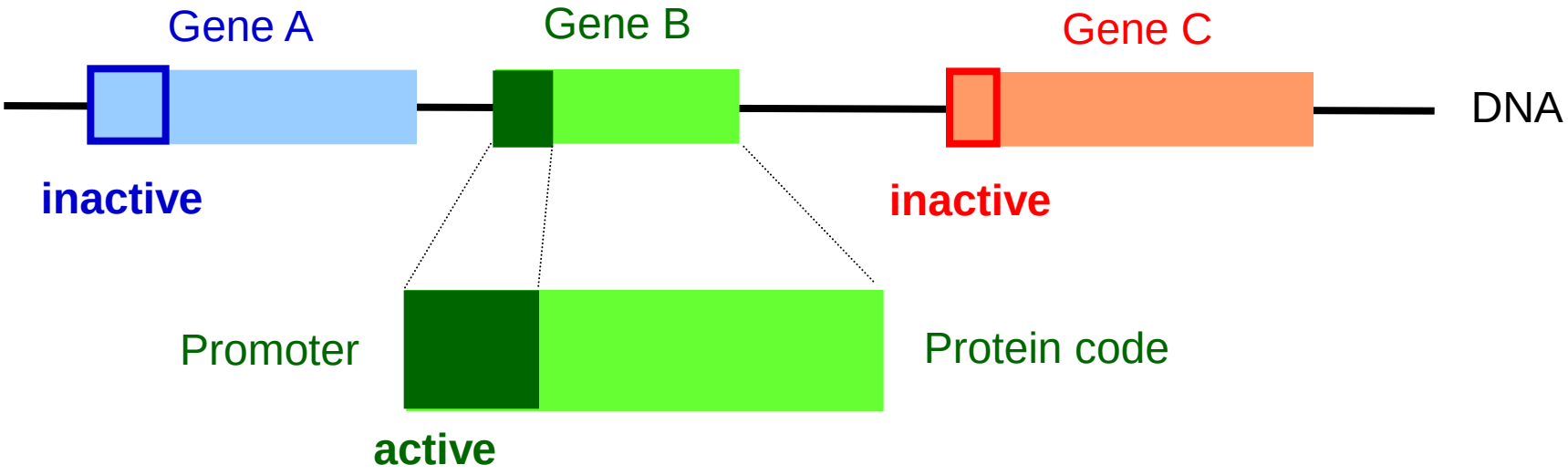
-small numbers of important molecules
-locality, gradients matter

Gene Activation/Repression Systems

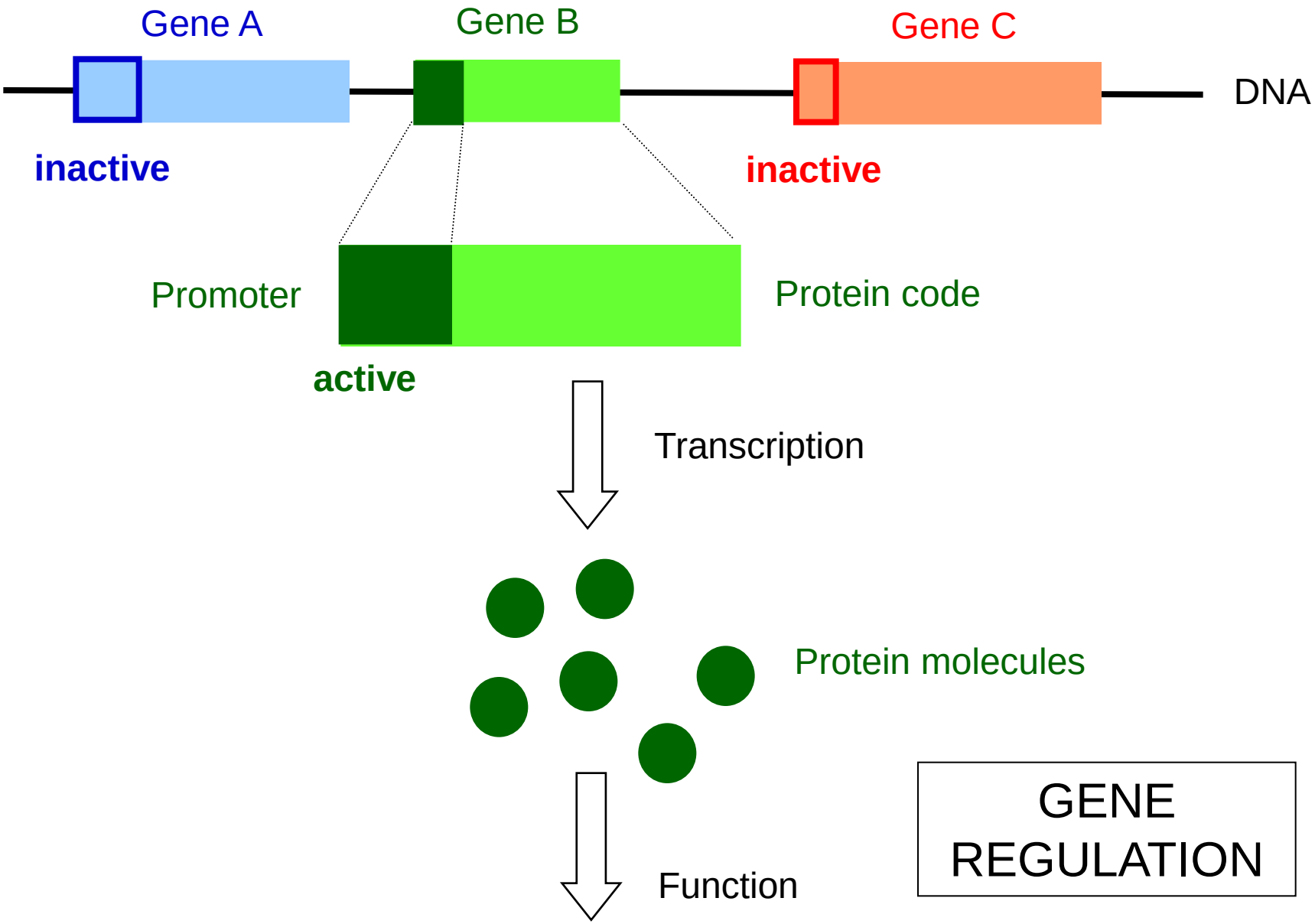


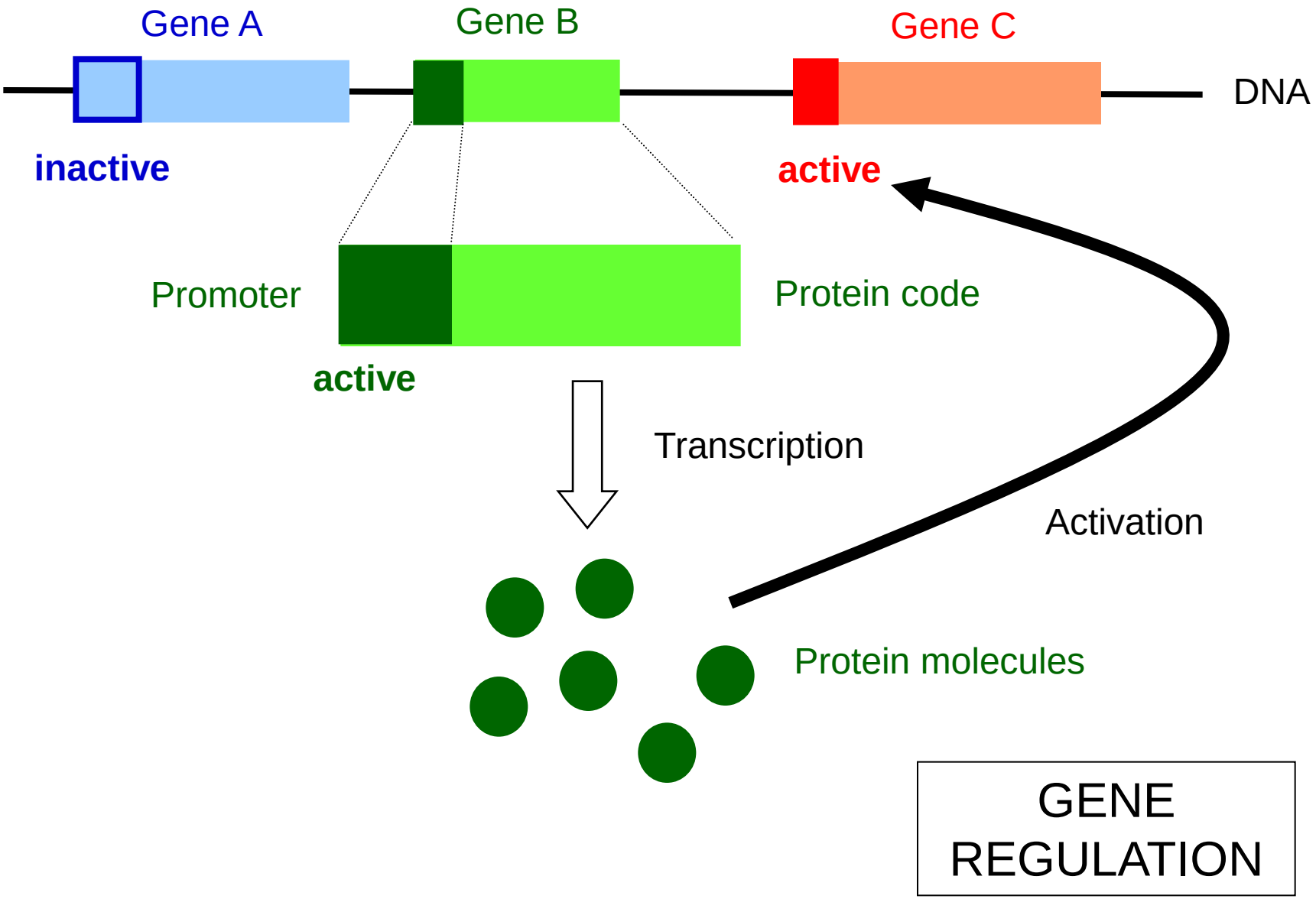
Rafiq K et al. Development 2012; 139:579-590.

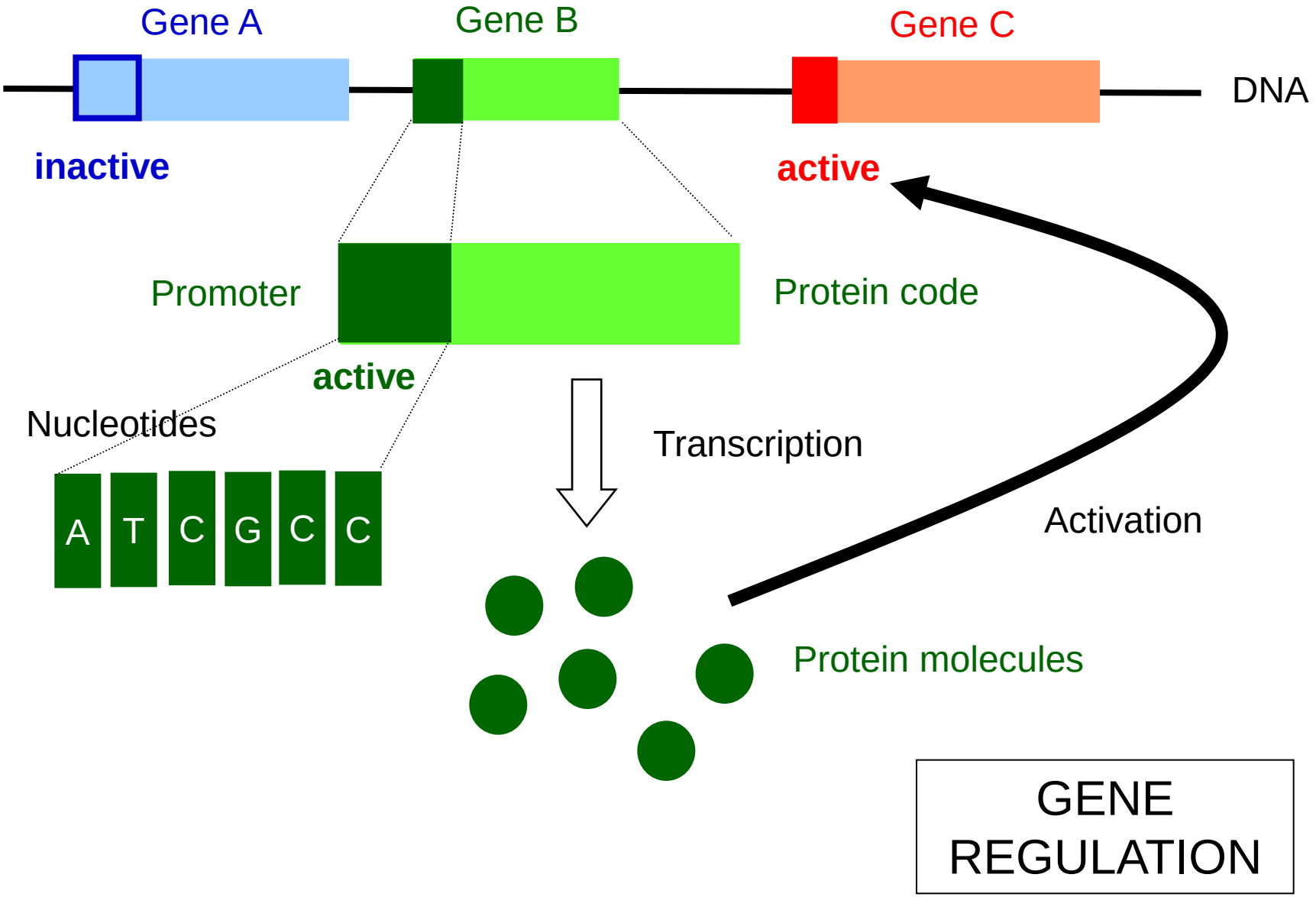

Gene A
Gene B
Gene C
DNA

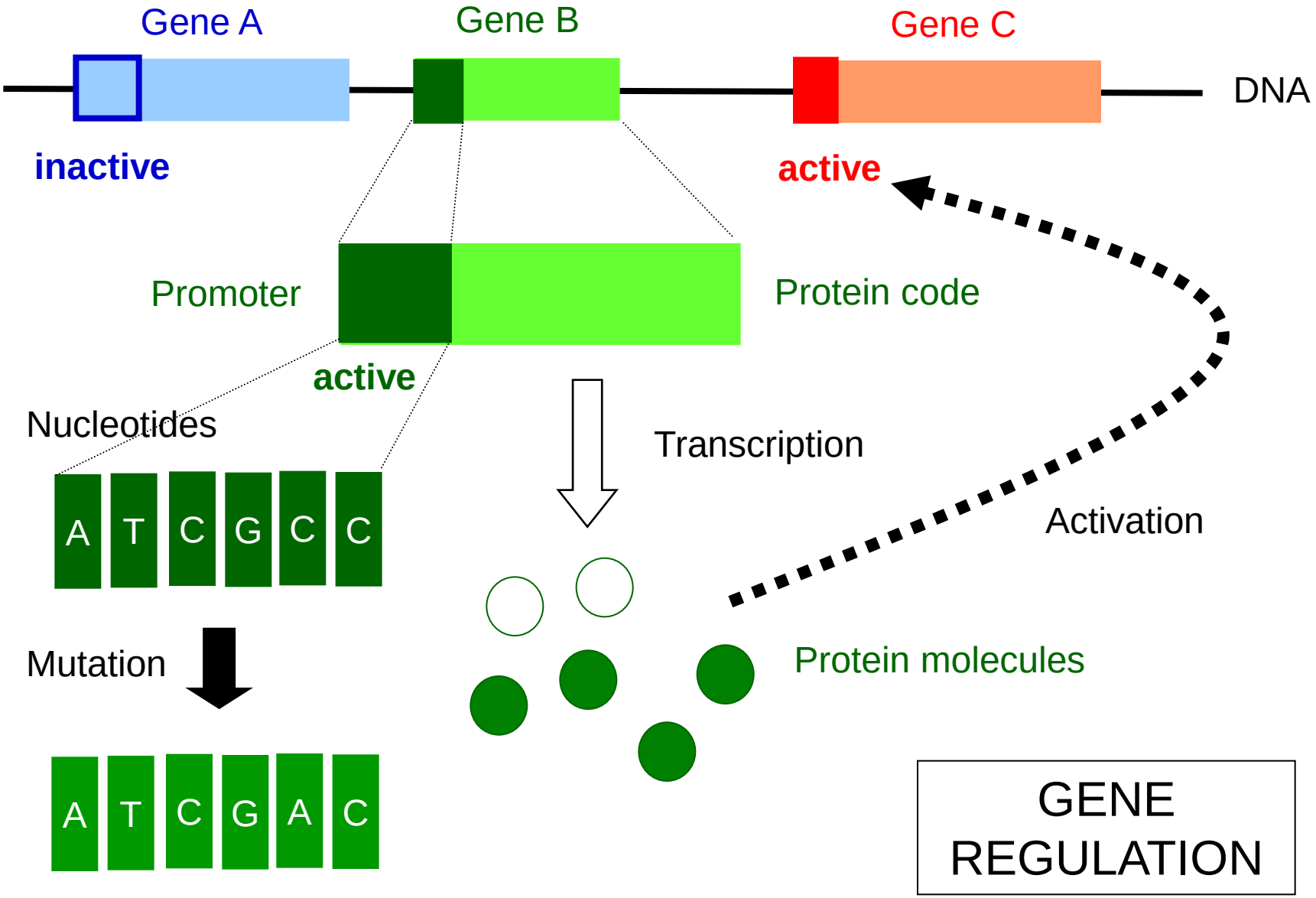


GENE REGULATION

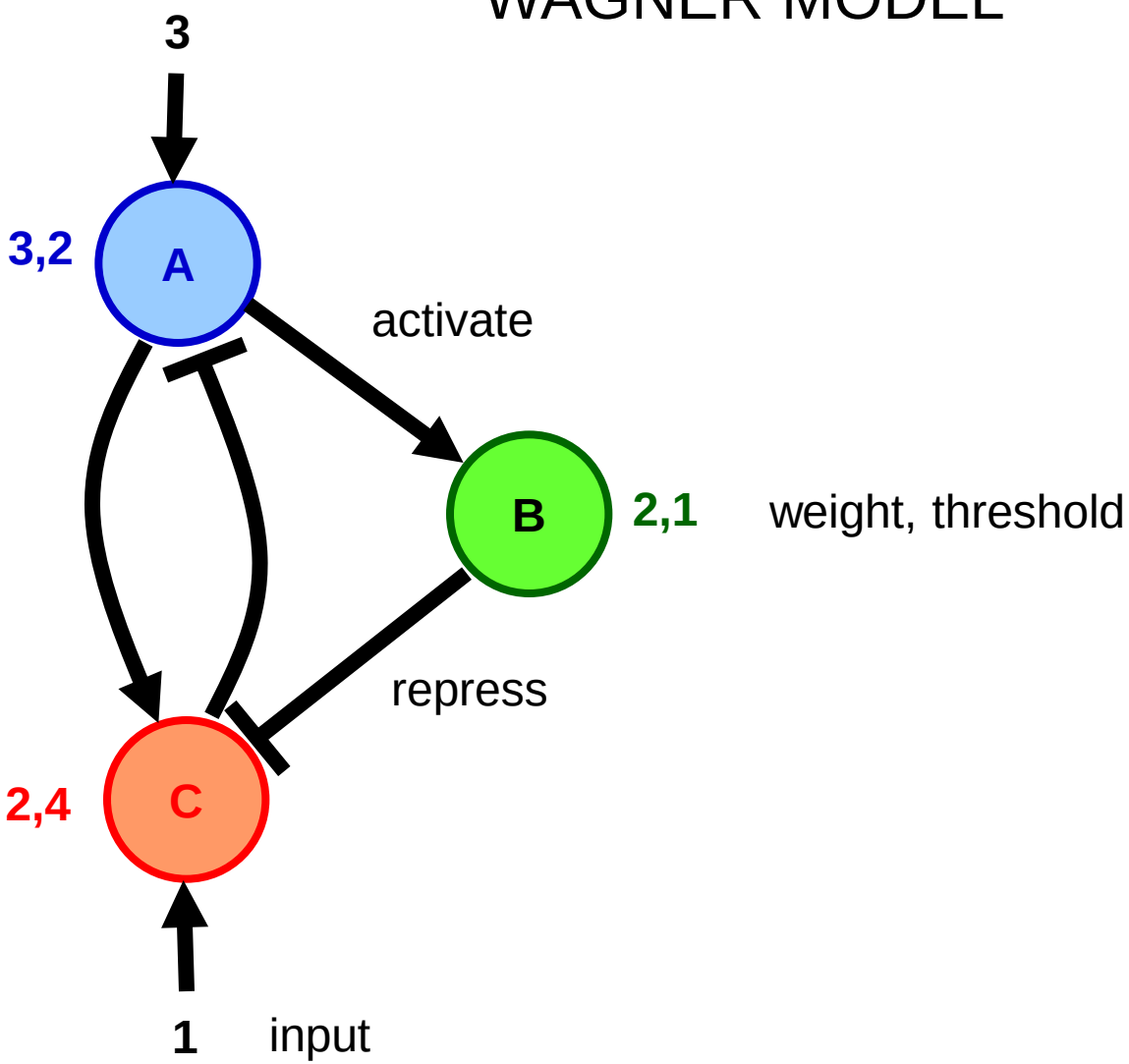

Gene A
Gene B
Gene C
DNA
inactive
inactive
Promoter
Protein code
active
Transcription
Protein molecules
Function
GENE REGULATION


Gene A
Gene B
Gene C
DNA
inactive
active
Promoter
Protein code
active
Transcription
Activation
Protein molecules
GENE REGULATION

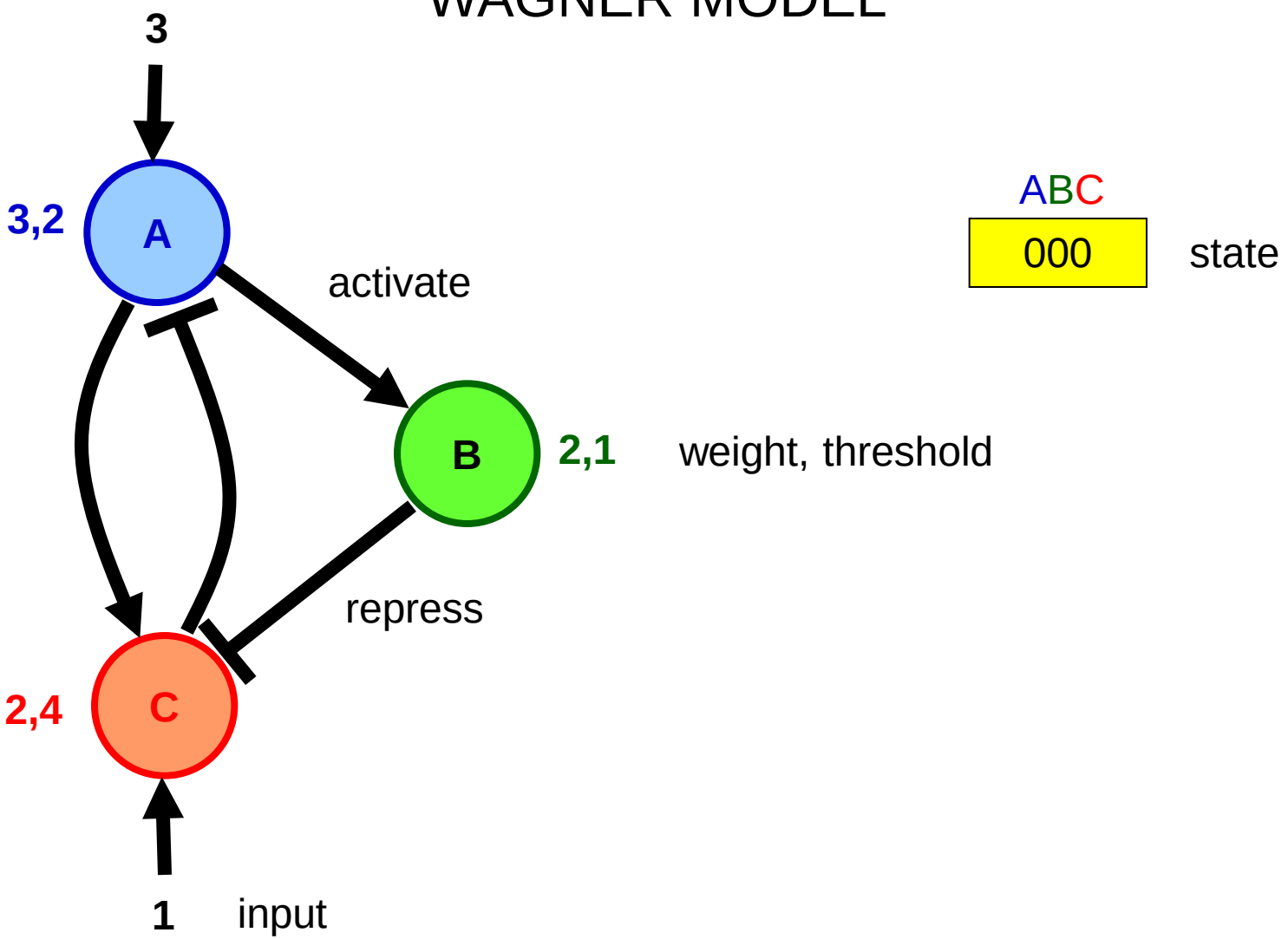

Gene A
Gene B
Gene C
DNA
inactive
active
Promoter
Protein code
active
Nucleotides
A T C G C C
Transcription
Activation
Protein molecules
GENE REGULATION


Gene A
Gene B
Gene C
DNA
inactive
active
Promoter
Protein code
active
Nucleotides
A T C G C C
Transcription
Activation
Mutation
Protein molecules
A T C G A C
GENE
REGULATION

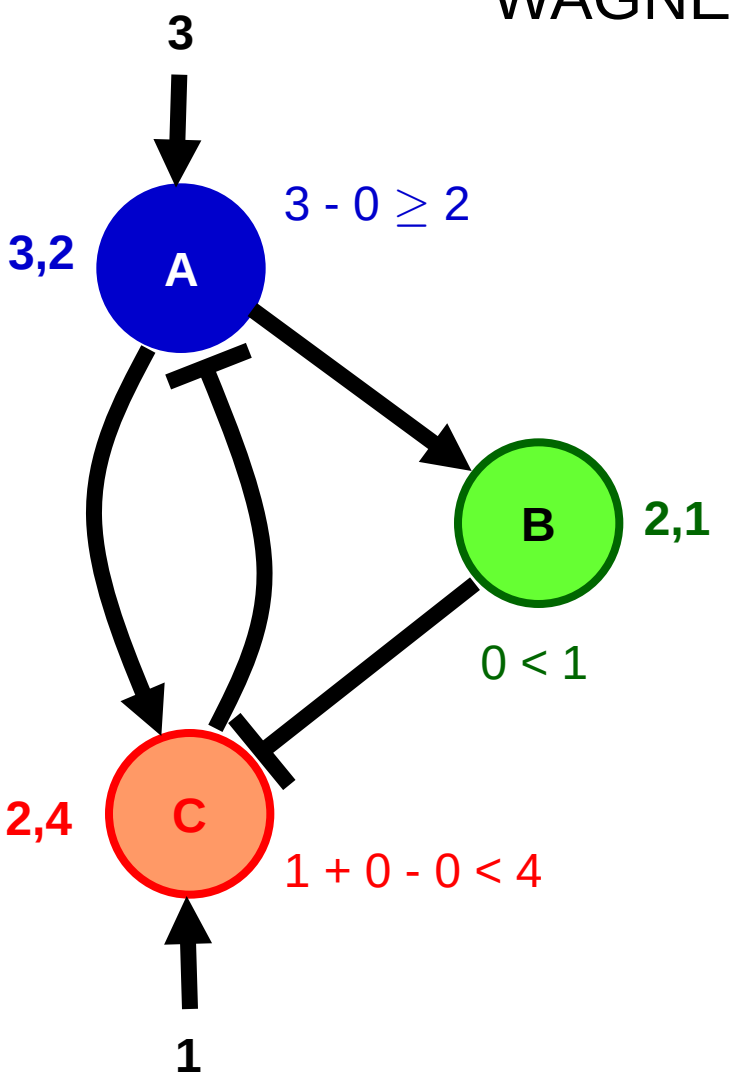
WAGNER MODEL

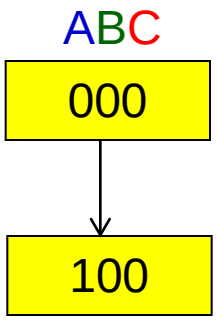


WAGNER MODEL

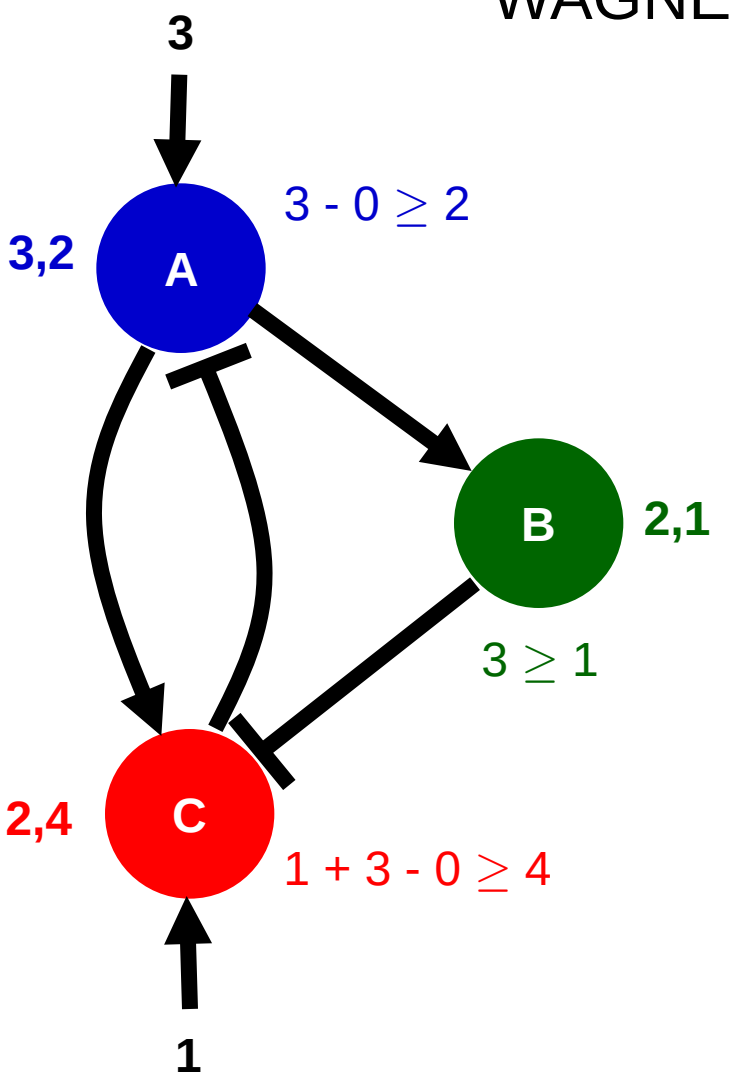


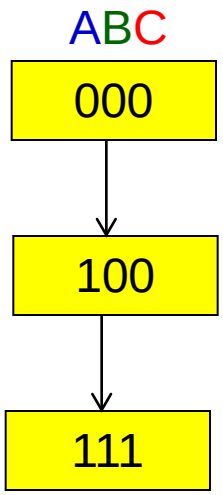
WAGNER MODEL



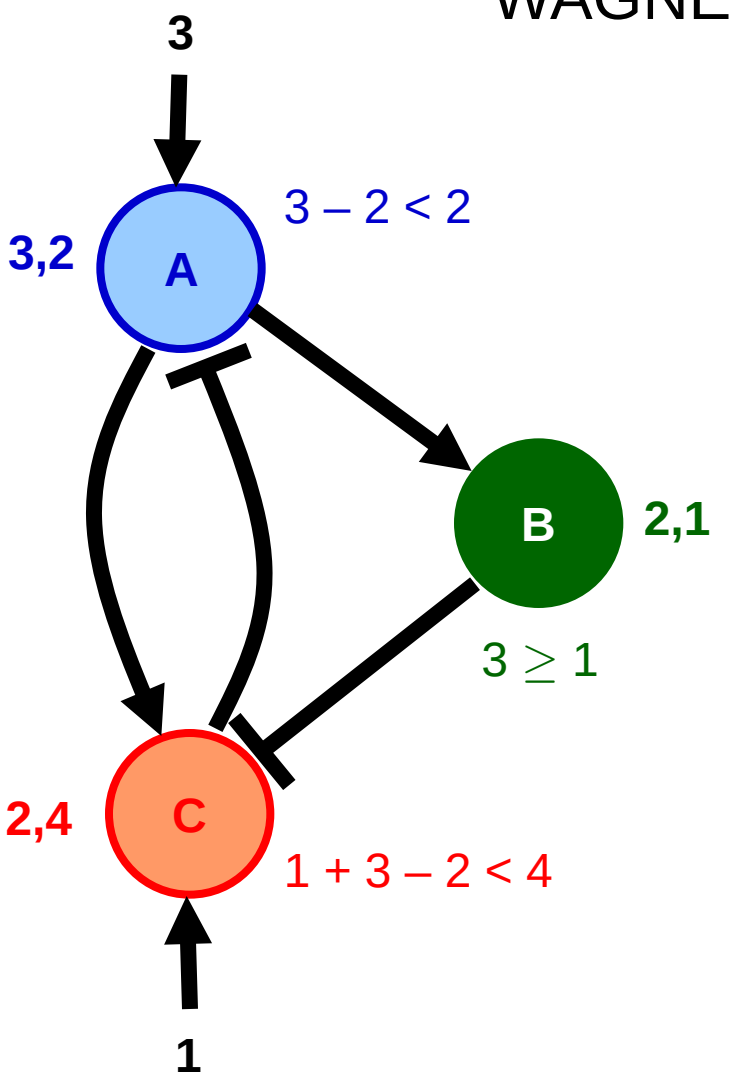


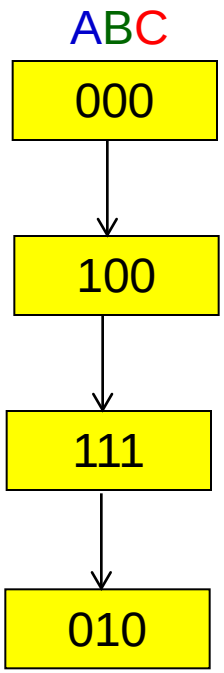
WAGNER MODEL





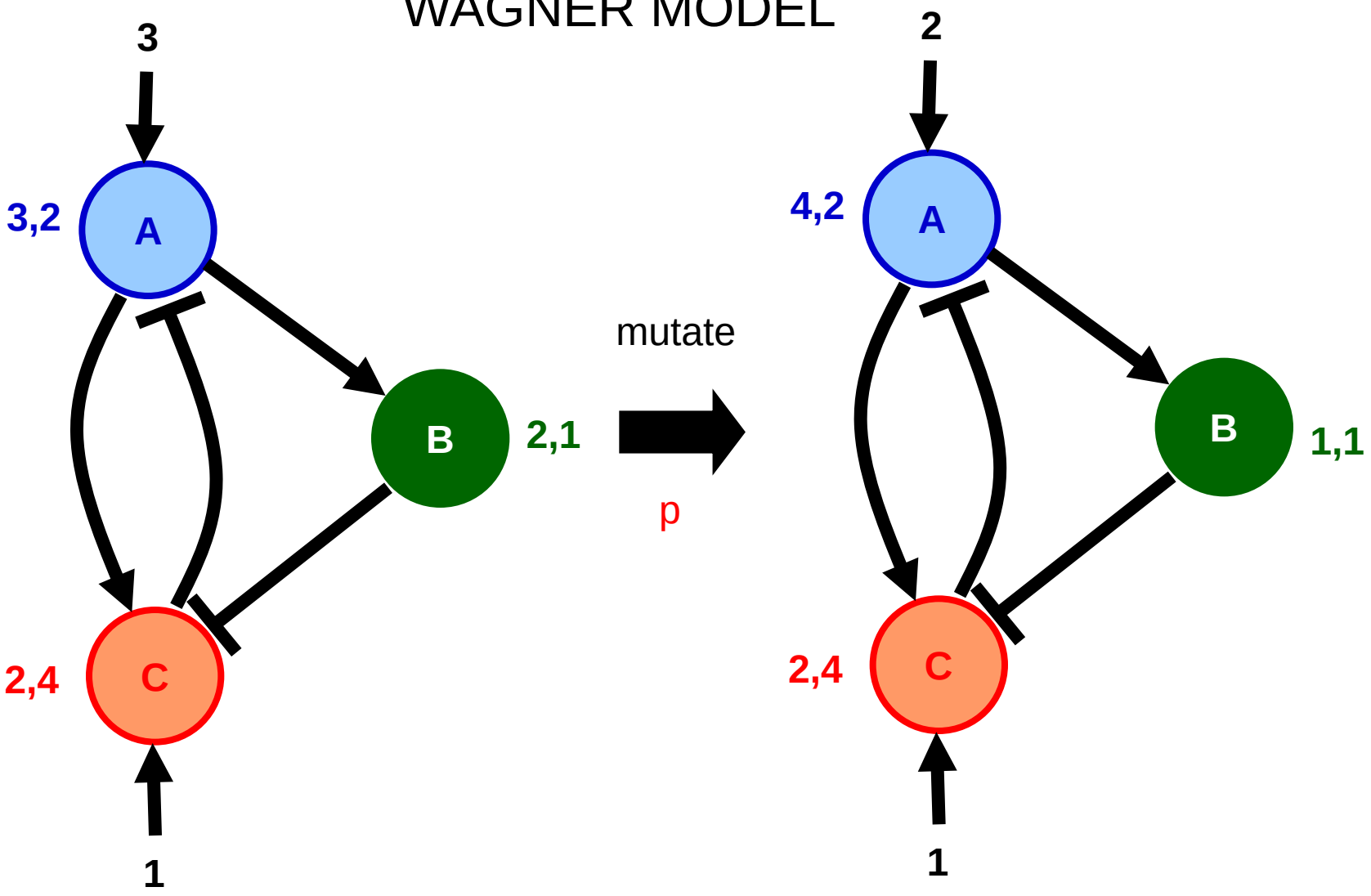
WAGNER MODEL





deterministic transition system

WAGNER MODEL

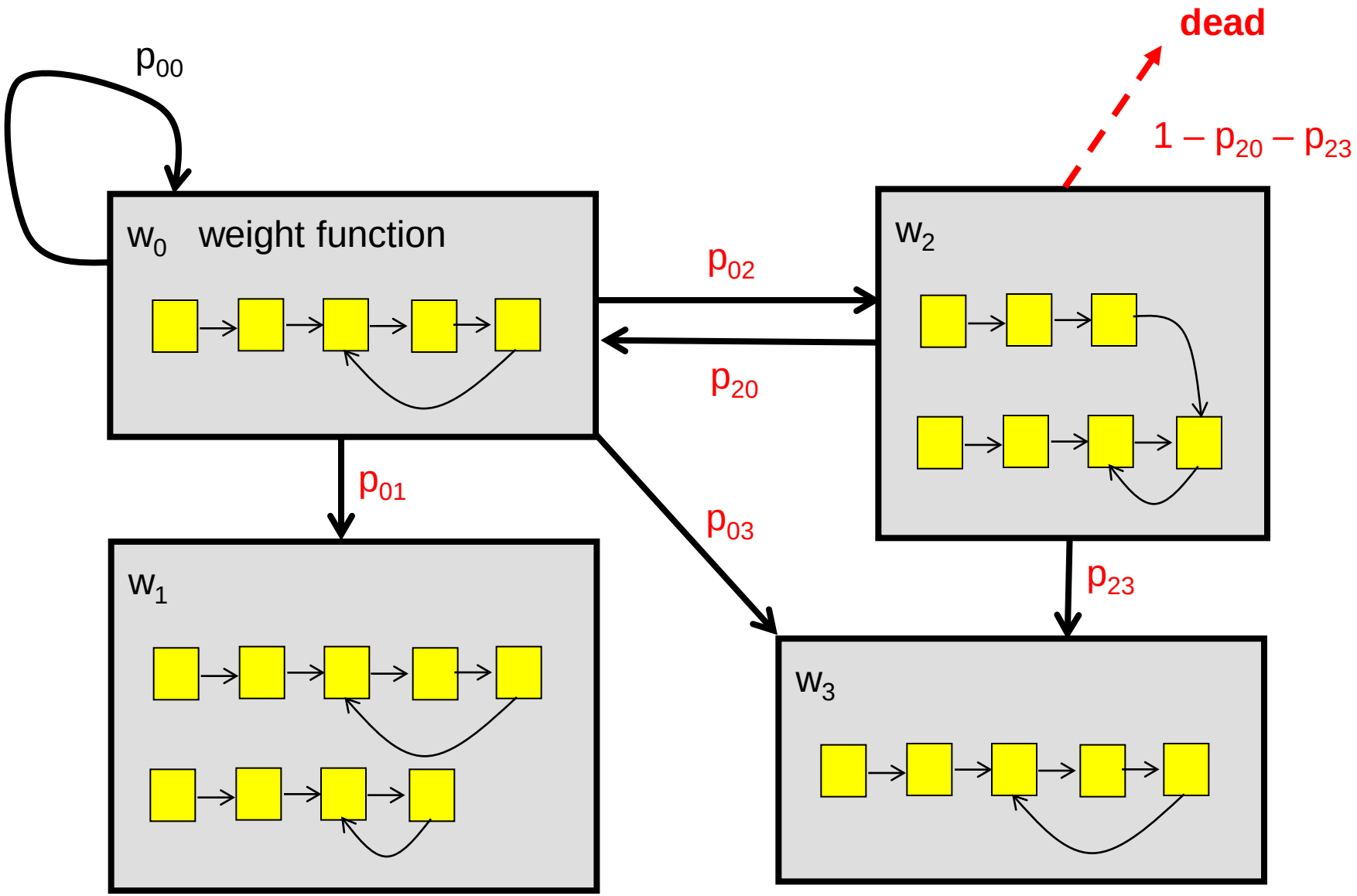


MUTATION PROBABILITIES

1. Each nucleotide (A, T, C, or G) mutates to another nucleotide with a given probability $\pi/3$.
2. The weight of a gene g decreases with the number of mutated nucleotides in the promoter region:

   if g has a promoter region of length n and k nucleotides are mutated, then $w(g)$ decreases to $w(g) \cdot (1 - k/n)$.

EVOLVING GENE REGULATORY NETWORK: DISCRETE-TIME MARKOV CHAIN (DTMC)



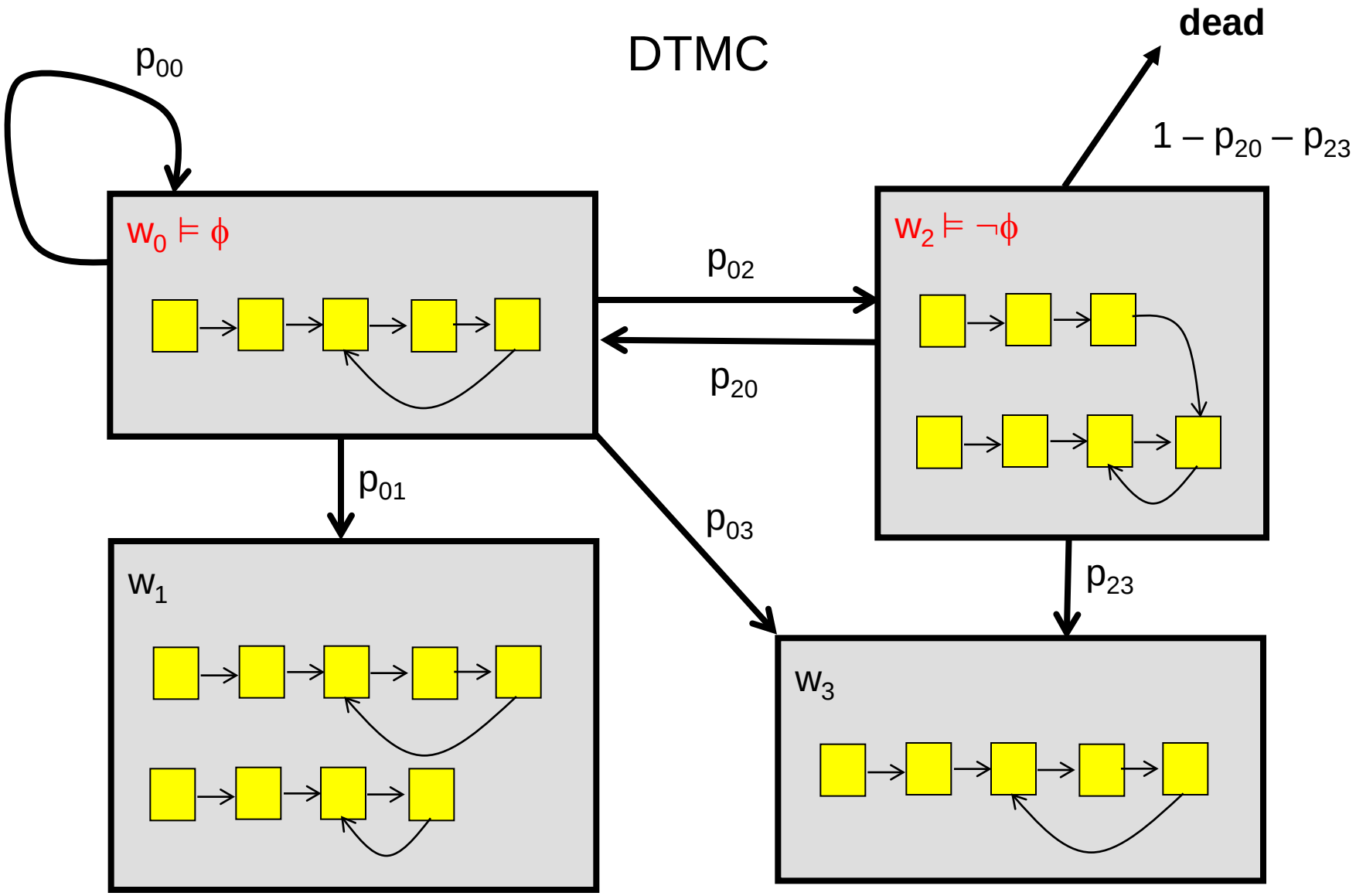
PHENOTYPE = TEMPORAL PROPERTY ϕ

Oscillation $\Box((A \Rightarrow \Diamond \neg A) \wedge (\neg A \Rightarrow \Diamond A))$

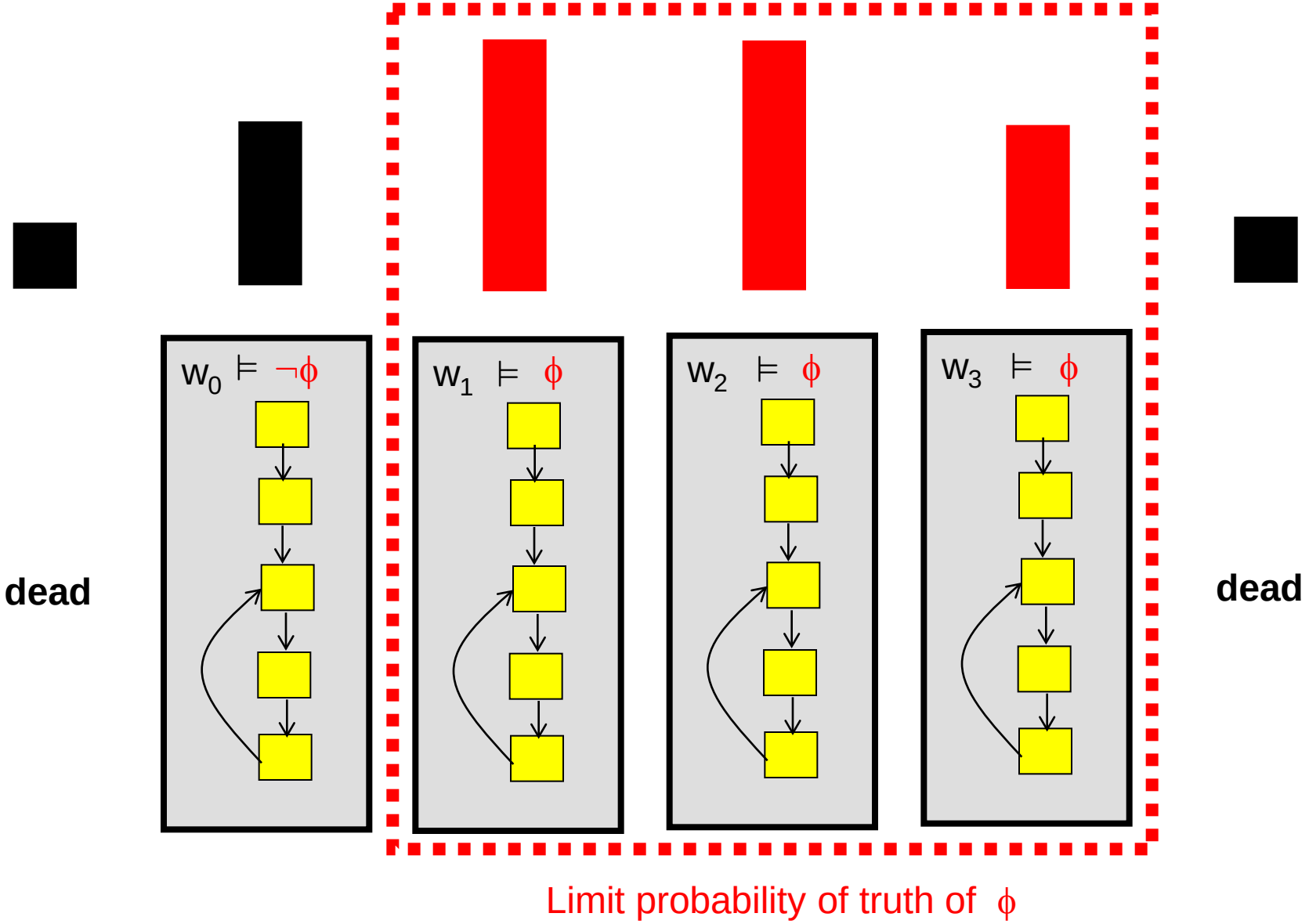
Bistability $\Box((A \wedge \neg B \Rightarrow \Box(A \wedge \neg B)) \wedge (\neg A \wedge B \Rightarrow \Box(\neg A \wedge B)))$

atomic propositions = genes

Elowitz & Leibler, *Nature* 2000.
Milo et al., *Science* 2002.


DTMC
dead
p_{00}
$1 - p_{20} - p_{23}$
$w_0 \models \phi$
$w_2 \models \neg\phi$
p_{02}
p_{20}
p_{01}
p_{03}
p_{23}
w_1
w_3

STATIONARY LIMIT DISTRIBUTION



COMPUTING THE ANSWER

1. Repeated Execution

Problem:

The number of weight functions w grows exponentially with the number of genes.

COMPUTING THE ANSWER

1. Repeated Execution

Problem:

The number of weight functions w grows exponentially with the number of genes.

Solution:

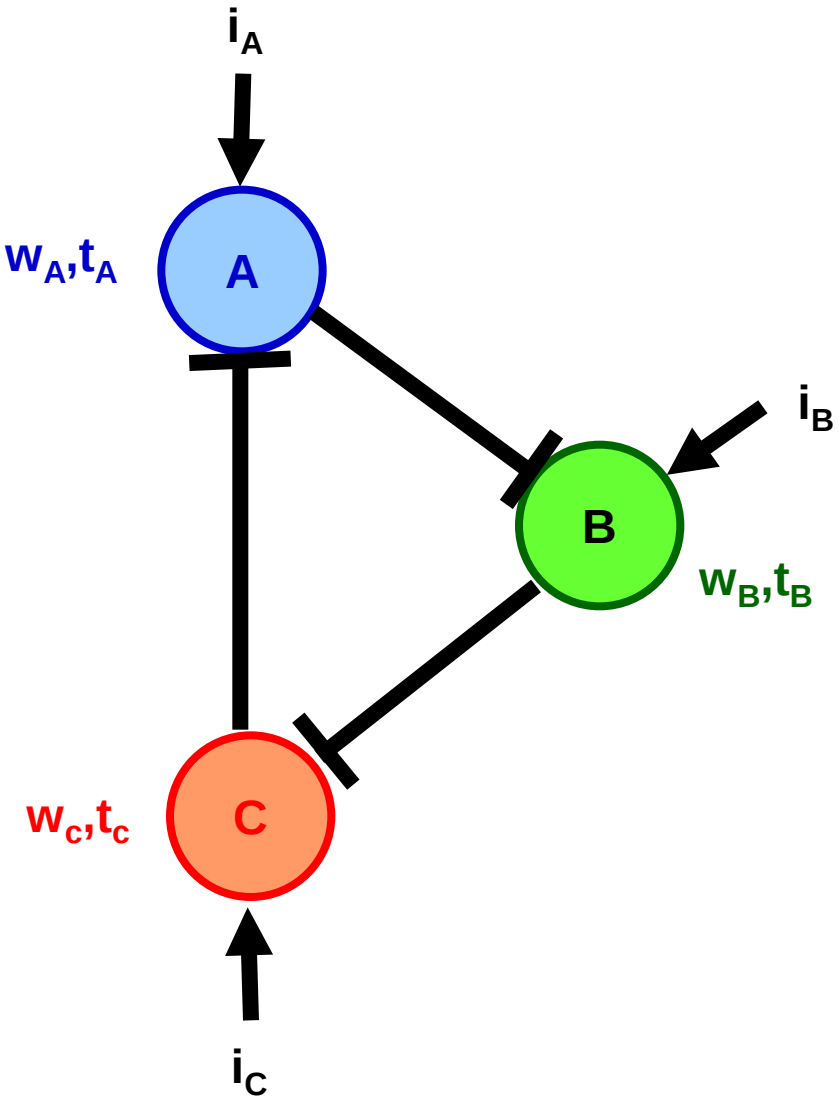
2. Parametric Model Checking

[Giacobbe, Guet, Gupta, H., Paixao, Petrov 2015]

PARAMETRIC MODEL CHECKING

1. Keep inputs, weights, and thresholds as symbolic values (“parameters”).

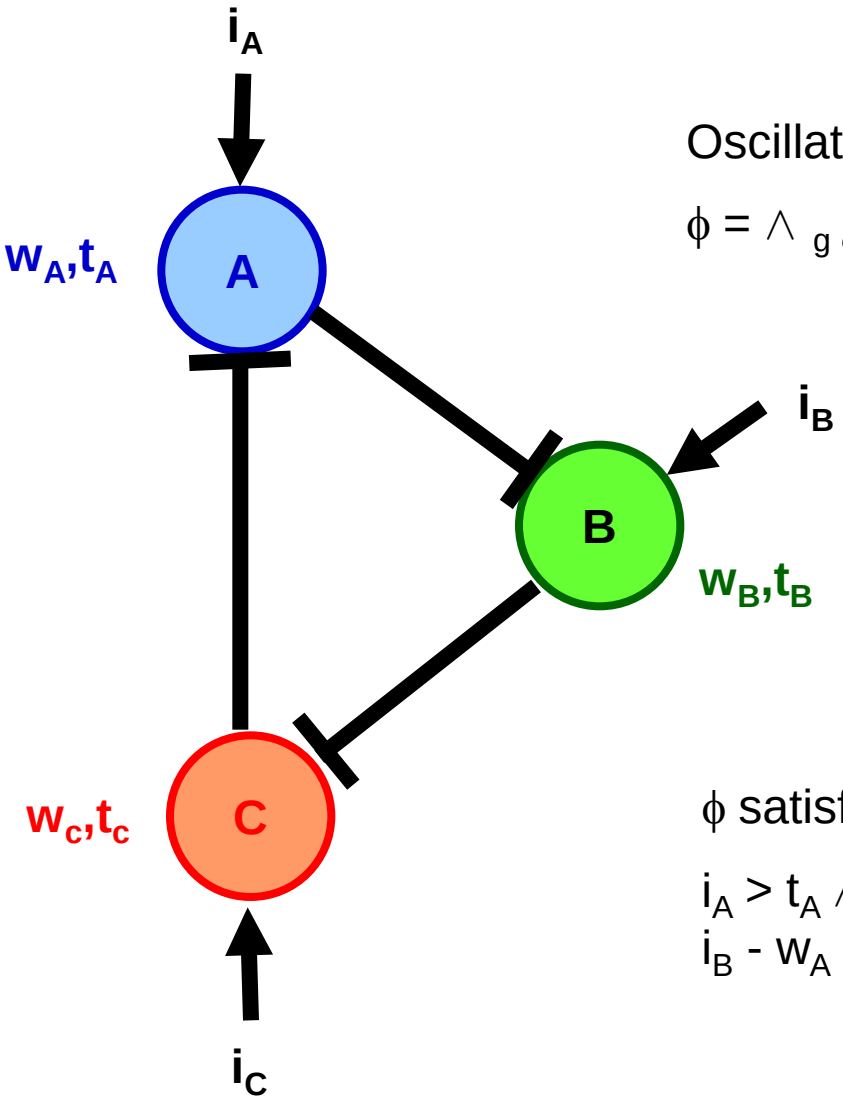
EXAMPLE: TRIPLE REPRESSOR



PARAMETRIC MODEL CHECKING

1. Keep inputs, weights, and thresholds as symbolic values (“parameters”).
2. Use SMT solving to compute the constraints on the parameter values such that an LTL formula ϕ is satisfied.

EXAMPLE: TRIPLE REPRESSOR



Oscillation:

$\phi = \wedge_{g \in \{A,B,C\}} \Box((g \Rightarrow \Diamond\neg g) \wedge (\neg g \Rightarrow \Diamond g))$

ϕ satisfied iff

$i_A > t_A \wedge i_B > t_B \wedge i_C > t_C \wedge$
$i_B - w_A < t_B \wedge i_C - w_B < t_C \wedge i_A - w_C < t_A$

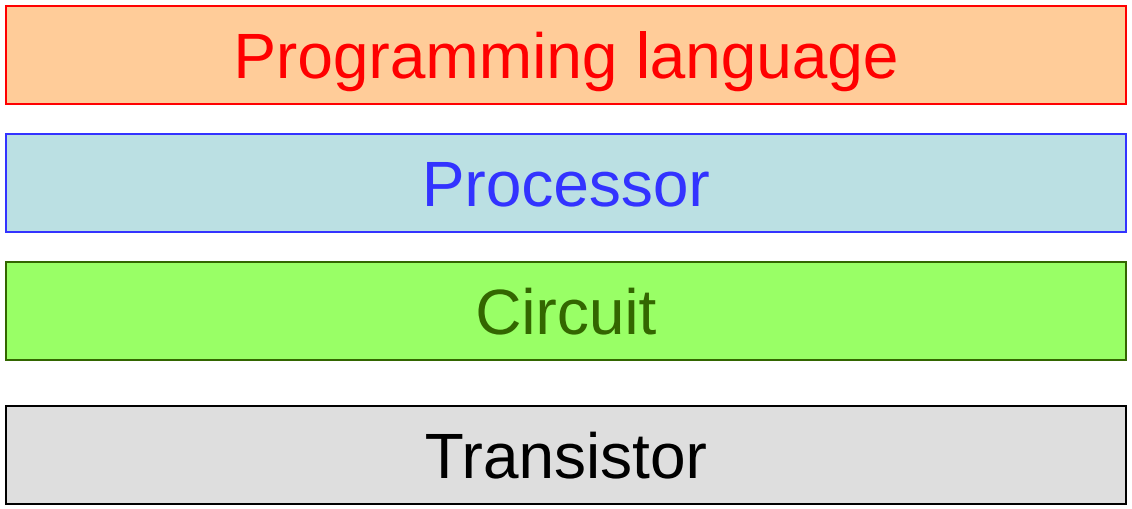
LIMITATIONS OF THE MODEL

-(over?)simplification of timing

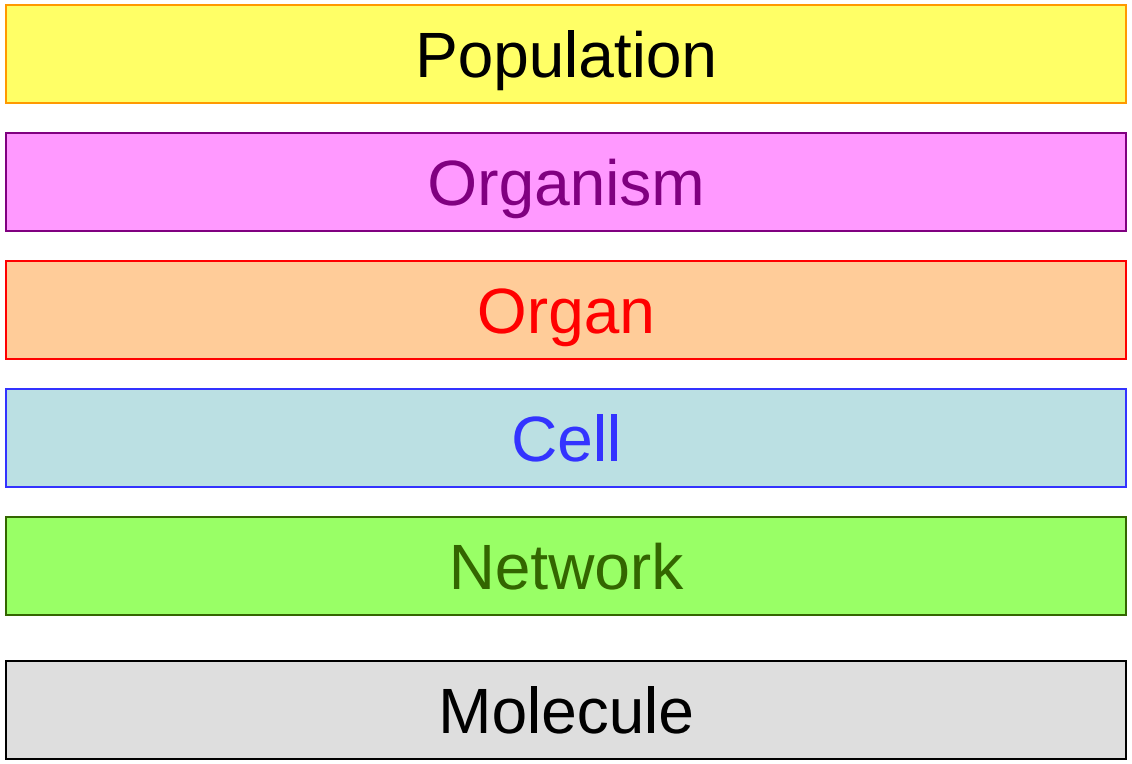
-(over?)simplification of quantitative measures (weights)

The Essence of Computer Science

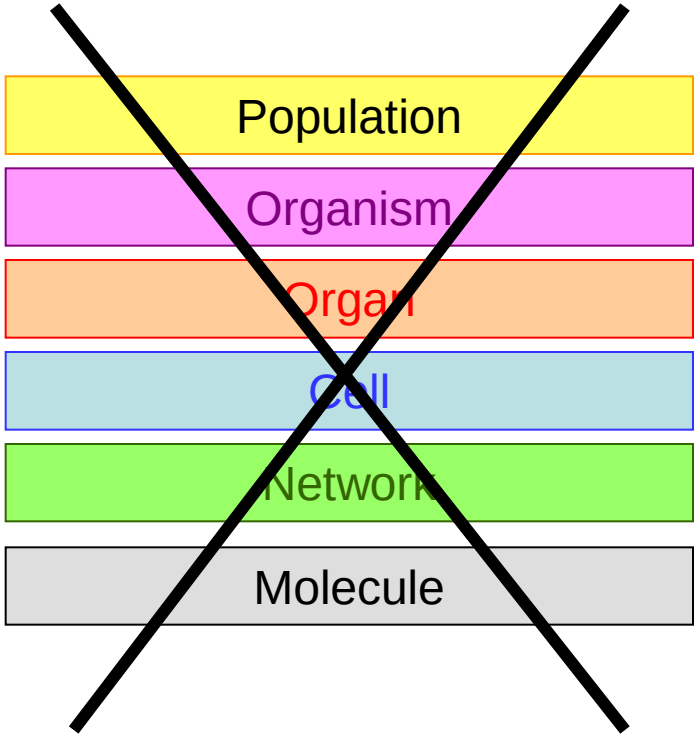
1. Algorithms
2. Machines/languages: towers of abstraction



Tower of Bio-Abstractions



Are there useful macros to structure the “hairball”?



We are looking for the organizing principle:
the programming language, only then for the program!